

Spacecraft Pose Estimation Using a Monocular Camera

by

Jian-Feng Shi

A thesis submitted to the Faculty of Graduate and Postdoctoral Affairs
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Aerospace Engineering

Carleton University
Ottawa, Ontario

© 2019

Jian-Feng Shi

This work is dedicated to my grandmother Wei-Yi, aunt Ze-Lin, and Ainney.

Abstract

Spacecraft pose estimation is an essential input to the Guidance, Navigation and Control process. Pose estimation based on monocular camera images does not require modifications to the target vehicle and has fewer resource requirements than LIDAR systems. We provide a comprehensive investigation and novel contributions in foreground recognition and extraction, image feature generation, and pose estimation. We evaluated 12 image feature detector and descriptor performances and proposed a new biologically inspired image descriptor. We also assessed the bag of visual words codebook technique for object localisation and evaluated linear, non-linear and non-parametric classifiers. We tested the ResNet and Inception-ResNet convolutional neural networks on target localisation. We developed compact semantic segmentation autoencoders based on AlexNet, U-Net and VGG. We made several new contributions in the image saliency generation. First, we developed a novel principal component analysis based formula for graph manifold ranking Optimal Affinity Matrix inversion which reduces computation time and stabilises the ranking inversion process. We developed a novel weighted gradient orientation histogram feature for monochromatic image superpixel identification and provided three enhanced versions of the graph manifold ranking tested on 32,536 images. Our technique out-performs the state-of-the-art saliency method in precision and our fastest method is $12\times$ faster than the original graph manifold ranking technique. We introduce an innovative false-coloured high-frequency salient feature image to enhance foreground and background pixel histogram distinction. We propose a novel space background classification scheme using pixel statistics to detect Earth passage. We evaluated appearance based pose matching using principal components analysis, SoftPOSIT and $ePnP$ for pose estimation. We propose a novel homography transform projection method that simplifies the perspective- n -point correspondence. We introduce improvements to the SoftPOSIT initiation to reduce the effects of local minimum trapping using centroid matching. We developed region-based pose estimation using level-set segmentation and pixel statistics. Our tests show the region-based method out-performs the appearance-based and point-based methods in speed, precision and stability.

Acknowledgements

I want to thank my friend and advisor Professor Steve Ulrich, for his constant trust, recommendations, and encouragements. I will always cherish the fond memories of our ride from Toronto to Ottawa where we brainstormed ideas for the future. My project would not be possible without his support and recommendations.

I want to thank Professor Karl Gerhard Roth for teaching me computer vision. Professor Roth mentored me technically and helped me to improve my writing. I am grateful for his teachings, inspirations, and encouragements.

I want to thank Mr Stéphane Ruel of Neptec Design Group Ltd. for his support, inspiration, motivation and recommendations. I want to thank Mr Andrew Allen of MDA, who have always supported my ideas, encouraged me and provided me with guidance. I want to thank Dr Chris Damaren who have guided, supported and encouraged me for so many years.

I want to thank Professor Junjie Gu for his motivation, advice, and recommendations. I want to thank Dr Kerman Buhariwala for being a great friend and mentor. I want to thank Ms Darlee Gerrard, Ms Jacqueline Maldonado, Mr Ozgur Gurtuna, Mr Eric Choi, Mrs Dawn Britton, and Dr Cameron Ower; this PhD would not be possible without their kind recommendations.

I like to thank Professor Jochen Lang, Professor Karl Gerhard Roth, Dr Chad English, Professor Jie Liu, and Professor Steve Ulrich for evaluating this work and provide valuable comments and suggestions.

This research was jointly funded by the Natural Sciences and Research Council (NSERC) of Canada's Alexander Graham Bell Canada Graduate Scholarship CGSD3-453738-2014, NSERC Engage grant with MDA (469958-14) and Neptec Design Group Ltd. (470485-14), Canadian Space Agency Space Technology Development Program, and Ontario Centres for Excellence Voucher for Innovation and Productivity II Award 24053. The scholarship awards and teaching assistantships from Carleton University's Department of Mechanical and Aerospace Engineering and the Office of Graduate Studies are also gratefully acknowledged.

Table of Contents

Abstract	iii
Acknowledgements	iv
List of Tables	x
List of Figures	xiii
List of Acronyms	xxvi
Chapter 1 Introduction	1
1.1 Background	1
1.1.1 ProxOps History and Technology	2
1.1.2 Pose Estimation	3
1.2 Motivation	7
1.3 Problem Statement	8
1.4 Literature Review	8
1.4.1 Target Recognition and Localisation	9
1.4.2 Target Extraction	11
1.4.3 Appearance-based Pose Estimation	15
1.4.4 Point-based Pose Estimation	16
1.4.5 Region-based Pose Estimation	19
1.5 Research Approach	21
1.6 Contributions	23
1.6.1 Target Extraction	23
1.6.2 Region-based Pose Estimation	25
1.6.3 Target Recognition and Localisation	26
1.6.4 Points-based Pose Estimation	26
1.6.5 Simulation Environment	27

1.7	Thesis Outline	27
Chapter 2	Image Processing and Feature Extraction	29
2.1	Image Processing	30
2.2	Edge Generation	32
2.2.1	Hough Transform	33
2.2.2	Point Inflation	35
2.3	Image Features	36
2.3.1	Feature Comparison	47
2.3.2	<i>y</i> BRIEF	51
Chapter 3	Target Localisation	65
3.1	Bag of Visual Words	65
3.1.1	Textons	67
3.1.2	Codebook Dictionary	68
3.1.3	Classifiers	75
3.1.4	Training and Inference	83
3.1.5	Localisation Performance	86
3.2	CNN Bounding Box	92
3.2.1	Overview	92
3.2.2	ResNet	93
3.2.3	Inception-ResNet	95
3.2.4	Object Detection	97
3.2.5	Metric Description	99
3.2.6	Localisation Performance	103
Chapter 4	Background Removal	110
4.1	CNN Segmentation	110
4.1.1	Network Description	110
4.1.2	Image Datasets	113
4.1.3	Evaluation Metrics	115

4.1.4	CNN Segmentation Performance	115
4.2	Real-time Saliency Extraction	117
4.2.1	Overview	121
4.2.2	Speed Improvements	121
4.2.3	Graph Manifold Ranking	122
4.2.4	Monochromatic Features	129
4.2.5	Image Texture from Feature Descriptors	133
4.2.6	Enhanced GMR	134
4.2.7	Dataset and Metrics	137
4.2.8	Foreground Extraction Performance	141
4.3	Infrared Spacecraft Image Saliency Extraction	147
4.3.1	Background Classification	149
4.3.2	Nadir Pointing	154
4.3.3	False-colour High-Frequency Saliency Feature Image	156
4.3.4	Algorithm Performance	159
Chapter 5	Appearance-based Methods	169
5.1	Principal Component Analysis	170
5.1.1	Dimensional Reduction	172
5.1.2	Low-rank Matrix Approximation	173
5.1.3	Inference	174
5.2	Appearance Matching	174
5.3	Pose Estimation Performance	176
5.3.1	Single-axis Pose Estimation	176
5.3.2	Multi-axis Pose Estimation	181
5.4	Image Occlusion Optimisation	182
5.4.1	Dissimilarity Measure	186
5.4.2	Pre-image Computation	186
5.4.3	ePCA Occlusion Optimisation Algorithm	189
5.4.4	ePCA Occlusion Optimisation Results	189

Chapter 6	Point-based Methods	193
6.1	<i>P</i> nP Method Overview	194
6.2	Internal 3D Model Generation	195
6.3	Camera Model	195
6.4	Feature Matching	199
6.5	Homography, RANSAC and Sentinel	200
6.6	Model Projection <i>P</i> nP	202
6.7	SoftPOSIT	203
6.7.1	SoftPOSIT Formulation	204
6.8	Enhanced SoftPOSIT	205
6.8.1	Global Minimum Search	206
6.8.2	Control Parameter based on Correspondence	208
6.8.3	Control Parameter based on Centroid Matching	208
6.8.4	Enhanced Performance	213
6.9	<i>eP</i> nP	216
6.10	Estimation Performance	223
Chapter 7	Region-based Method	228
7.1	PWP3D	228
7.1.1	Notation Definitions and Rotation Transformations	229
7.1.2	Level-set Pose Estimation	230
7.2	Region-based Method Enhancements	235
7.2.1	Centre Initialisation	235
7.2.2	Gradient Descent	236
7.3	Estimation Results	238
7.3.1	Stepsize Sequence	238
7.3.2	A Priori Enhancement	240
7.3.3	Pose Estimation Error	241
7.3.4	Envisat 6-DOF Pose Estimation	245
7.3.5	Reduced Model Pose Estimation	246

7.3.6	ISS Pose Estimation	249
Chapter 8	Conclusion	259
8.1	Summary	259
8.2	Future Work and Recommendations	261
	Bibliography	262
	List of Videos	295
	Biographical Sketch	322

List of Tables

2.1	List of image feature models used for comparison. The code number represent the year of publication.	39
2.2	Oxford dataset descriptions.	48
3.1	Comparison of keypoint precision using different classification techniques and texton label similarity tolerance. Results are computed based on an ISS thermal camera image, keypoint classification is performed manually. Hyper parameters are set as follows: 50 keypoints; no texton image PCA; feature vectors used in FD, HK-0, BO, and k NN-2, are converted to 10 principal component eigenspace. No PCA was applied to the NB method. The unique codebook of 595 textons was used as BoVW basis.	87
3.2	Comparison of keypoint precision using various classification techniques and number of keypoints. Hyper paramters: similarity tolerance is 15; no texton image PCA; feature vectors used in FD, HK-0, BO, and k NN-2, are converted to 10 principal component eigenspace. No PCA was applied to the NB method. The unique codebook of 595 textons was used as BoVW basis. Computation performed on Intel® Core™ 2 Quad Q6600 – 2. 4GHz Processor running on 32bit Windows-Vista-SP2. MATLAB 7.0 R14 was used for coding.	87
3.3	Comparison of keypoint precision using different classification techniques and codebook size. Hyper paramters: similarity tolerance is 15; 50 keypoints; no texton image PCA; feature vectors used in FD, HK-0 – 2, BO, and k NN-0 – 2, ∞ , inf , θ , are converted to 10 principal component eigenspace. No PCA was applied to the NB method.	90
3.4	Mean Jaccard index comparison.	90
3.5	Virtual camera parameters, HFOV is Horizontal Field of View.	101
3.6	SRCL spacecraft platform parameters.	102

3.7	Network performance for Class 1 objects. All results are computed using the inference datasets for synthetic and lab images. Comparing synthetic and real images and various network types. Detection Probability and IoU thresholds set to 0.5	107
3.8	Network performance for Class 1 objects. Comparing various network types with threshold variation. \overline{IoU} denotes average IoU over test images omitting images with $IoU = 0$	109
4.1	Network architectures; the first three numbers represent height, width, and channel of each feature layer, -sign indicates Batch Normalisation (BN) and Leaky ReLU (LR) combination layers. +sign indicate BN, LR and Drop Out (DO) combination layers. The value after ‘c’ indicates square kernel size, the value after ‘s’ indicates the stride, the value after ‘w’ indicates the window size, FC indicates Fully Connected, LRN indicates Local Response Normalisation.	114
4.2	Classification results, +sign indicate dropout added to convolutional layers.	116
4.3	Forward inference timing and semantic segmentation metric comparisons. Timing is an average of all evaluation images, semantic segmentation metric is based on the IR and the photo ISS images only.	118
4.4	Image datasets. NOTE:starred references (*) are where the datasets were downloaded from.	138
4.5	SatSeg Description	138
4.6	List of benchmark models used for comparison. NOTE: starred references (*) is where source code or executable was downloaded from.	139
4.7	Computing Platforms	141
5.1	Variations in number of training frames.	178

6.1	Descriptions of the model shapes, number of points per model, and β_0 initial values.	213
6.2	3D model IC orientation, all angles are in degrees.	214
6.3	Target object position IC, in length unit [L].	214
6.4	Target object orientation IC, all angles in degrees.	214
6.5	Pose estimation error success criteria.	214
6.6	SoftPOSIT enhancement study pose estimation run settings.	215
6.7	β_0 initialisation by centroid matching results.	215

List of Figures

1.1	<i>TriDAR</i> installed on Orbital Sciences Cygnus spacecraft. Courtesy of Neptec Design Group Ltd.	6
1.2	Canny edge lines in white, Harris corners in magenta circles, and SIFT keypoints in yellow crosses. When viewed with a non-illuminated space background in the case of the Envisat, the keypoints and edges are invariant inputs for pose estimation and tracking. Line and keypoint features require robust classification when there is a cluttered background.	19
1.3	Thesis topics organisation chart; solid lines indicate the pose estimation process; dashed lines indicate the investigated method category. Numbers represents section numbers, A.B represent Appendix B.	28
2.1	Image processing method comparisons. Bands in HER is due to rectangular grids and is smoothed linearly. Since the grid dimensions are known, it can be removed from subsequent edge detection.	32
2.2	Edge detection of a synthetic CubeSat image.	33
2.3	Combined edge histogram and response map.	34
2.4	Hough-space representation. The top voted points are circled.	35
2.5	Hough Transform straight edges overlay on top of the combined edge response map. The purple line represents the straight edge receiving the highest number of HT votes.	36

2.6	Edge detection of a difficult to process thermal image. The image contains a heat source on the upper left corner sending heated air towards the center of the image. A model spacecraft is in the lower-mid right side of the image. The first image on the left is the original image; the centre image shows Canny edges extracted from the raw input image; on the right is the HT edge computed from HER and combined edge generation techniques. The third image on the right shows points extracted from the spacecraft model where as the middle image shows only edges from the heat source is detected.	37
2.7	Inflating points along the HT straight lines.	37
2.8	Low resolution CubeSat PnP test. Compare corners verticies with edge inflated intermediate points.	38
2.9	SIFT features of the ISS thermal image. Each circle represents a keypoint position; the lines in the circles represent keypoint orientation, and the circle size represents the keypoint scale.	41
2.10	Oxford dataset images.	48
2.11	Image feature evaluation matrix.	49
2.12	Feature comparision results.	50
2.13	Human retina receptor density distribution. Figure reproduced from [1].	51
2.14	Random distribution histograms.	52
2.15	2D binary patterns.	52
2.16	yBRIEF test patterns: (o-Start, ×-End).	54
2.17	Lc-Differencing Patterns: (×-Cones,o-Rods ,△-Lc Points)	56
2.18	yBRIEF test images, Oxford image data set plus Iguazu. Top-left: Graffiti, top-right: Boat, bottom-left: Iguazu, bottom-right: UBC.	57
2.19	Thermal camera and test image.	58
2.20	Recall vs. 1-precision pattern performance.	59
2.21	Recall vs. 1-precision patch smoothing.	60
2.22	Recall vs. 1-precision for all features.	61

2.23	Match precision comparisons.	62
2.24	True match comparisons.	63
2.25	Feature matching comparison at 135 deg.	64
3.1	Object recognition training and inference process.	66
3.2	S+MR8 filter set with 13 S filters and 38 MR filters. Out of the 38 MR filters only 8 maximal response anisotropic maps are saved by collapsing the 6 orientations into a single response map.	70
3.3	Image patch clustering. Blank columns represent the original image patches and shaded columns represent clustered mean images. Key-point ID of each image patch is stored with each clustered group.	72
3.4	Solar panel image texton example. The solar panel image is processed for SIFT keypoints. Only the keypoints inside the boundary (red broken lines) are used. Image patches around the keypoints are extracted. Image patch are clustered using the k -Means method. Index 1 – 19 – 23 forms the first texton, 2 – 3 – 6 forms the second texton, and 14 – 16 forms the third texton.	73
3.5	S-MR8 filtered response maps for solar panel keypoint patches. Filter response map with the same colour shading belongs to the same image group from Fig. 3.4.	74
3.6	Local region bins; crosses represents the central keypoint, shaded circles represents keypoints that are assigned to the local region bin.	74
3.7	Example of a codebook forming a object feature vector.	76
3.8	ISS station module training images used for component recognition training.	84
3.9	ISS solar panel training images used for component recognition training.	84
3.10	ISS truss segment training images used for component recognition training.	85

3.11	SIFT keypoint and patches on the ISS infrared image. The keypoints in (a) is before NMS and random keypoint selection. The patches in (b) is after NMS and random keypoint selection reducing the number of keypoints to 30.	88
3.12	Unfiltered image patches with 32 pixel sides. 50 principal components were kept and converted back to the image domain.	89
3.13	Bounding boxes of the ISS query image. Subfigure (a) shows over prediction of the ISS modules. Subfigure (b) shows correct prediction of the ISS truss.	91
3.14	Simulation process including offline training. During inference, the dotted arrow indicates initialisation. For laboratory experiments, replace the <i>Synthetic Image Generation</i> block with real-world camera images.	94
3.15	Two-layer network with <i>shortcut connection</i> to drive residual learning [2].	95
3.16	101-Layer <i>ResNet</i> [2]. Red boxes represent filter kernels, and non-red boxes represent filter response maps and the input. The green arrows represent inputs <i>skip across</i> the residual responses. The gray box represent FC layers, and the green layer is the <i>softmax</i> classifier.	96
3.17	GoogLeNet Inception Module [3]. Red boxes represent filter kernels, brown boxes represent max pooling, and non-red boxes represent filter response maps.	97
3.18	The <i>Inception-ResNet-V2 Inception-ResNet-B</i> Module [4]. Red boxes represent filter kernels, and non-red boxes represent filter response maps.	97
3.19	Faster-RCNN Region Proposal Network.	100
3.20	3D Studio Max [®] environment for synthetic image generation.	101
3.21	Carleton University Spacecraft Robotics and Control Laboratory.	102

3.22	Simulated CubeSat detection using faster-RCNN based on various classifier networks. Subfigure (a) is using the <i>Inception-ResNet-V2</i> network and Subfigure (b) is using the <i>ResNet-101</i> network. https://youtu.be/AfBw4jGBz6Y	104
3.23	CubeSat test platform detection using faster-RCNN using different classifier networks. Subfigure (a) is using the <i>Inception-ResNet-V2</i> network and Subfigure (b) is using the <i>ResNet-101</i> network. https://youtu.be/M1ceJWM4pKM	105
3.24	Laboratory environment CubeSat platform detection using faster-RCNN based on various classifier networks. Subfigure (a) is baselined in <i>Inception-ResNet-V2</i> and Subfigure (b) is baselined in <i>ResNet-101</i> . https://youtu.be/GwaIWk6cjzs	106
3.25	Network performance for Class 1 objects. All results are computed using the inference datasets for synthetic and lab images; SYN is the synthetic CAD images from the CubeSat ProxOps simulation; REAL is the experimental lab images of the spacecraft platforms; IR is the <i>Inception-ResNet-V2</i> network model; R is the <i>ResNet-101</i> network model; <i>Estm Prob Lmt</i> is the probability threshold of 0.5, detection box with probability lower than this threshold is rejected; <i>Actual BBX Lmt</i> is the IoU threshold of 0.5, bounding boxes with IoU higher than this threshold are considered as Actual Positives (<i>AP</i>). Subfigure (a) provides IoU for various test conditions and network configurations. Subfigure (b) compares precision versus recall for <i>Inception-ResNet-V2</i> and <i>ResNet-101</i> models by varying both detection probability and IoU thresholds. Subfigure (c) and (d) compares network accuracy of varying detection probability and IoU thresholds. When computing accuracy, the non-varying threshold is held to the default value of 0.5 for both IoU and detection probability. Graphs are best viewed in colour.	108

4.1	Training and evaluation pipeline, refer to Table 4.1 for encoding network details. Decoding network mirrors the encoding network. Phase 1 network is trained for single class identification. Phase 2 transfers the network weights and batch normalisation parameters from phase 1. The encoder network parameters and weights are frozen during phase 2 training. A batch of 128 images forms the input tensor and is fed through the FCN with labelled target masks. Feature map tensors from the first convolutional layer of every encoder network groups are <i>skipped</i> to the respective decoder side. During inference, the FCN runs forward to generate the semantic segmentation map.	111
4.2	Sample images from the Space-5 database.	115
4.3	Semantic segmentation of the ISS. Top: IR camera image, bottom: photo camera image. Left to right: original, ground truth <i>AlexNet-8</i> , <i>UNet-8</i> and <i>VGG-19</i>	118
4.4	Background subtraction and semantic segmentation results, images are taken from STS-135 mission undocking sequence with background Earth motion. Equal partitioned frames are selected: 1, 55, 110, 164, and 219. Method as follows: level 1-original , level 2-manual , level 3-MOG [5] , level 4-MOG2 [6] , level 5-KNN [7] , level 6-GMG [8] , level 7-ADP [9] , level 8- <i>AlexNet-8</i> , level 9- <i>UNet-8</i> , level 10- <i>VGG-19</i> . MOG, MOG2, KNN, and ADP methods requires initialisation on the first frame, GMG method requires 120 frames to initialise.	119
4.5	Superpixel breakdown of an infrared ISS image. Subfigures 4.5(a) to 4.5(c) shows LSC, SLIC, and SEEDS methods respectively. Red circles indicate locations of the superpixel spatial centroid.	123

4.6	Superpixel performance comparisons. The first two columns indicate timing ratios dividing LSC and SLIC timing for 70 and 300 superpixels respectively. Columns 3 to 6 provides Mean F -measure, Max F -measure, AUC, and MAE.	123
4.7	OAM reduction results. Each data point is the average of all 32,536 image calculations. Threshold level corresponds to Eq.(4.10). Figure 4.7(a) shows the timing ratio between reduced OAM over non-reduced for a single OAM calculation. Figure 4.7(b) shows the average-relative-percent-error in the reduced OAM. Subfigure 4.7(c) and 4.7(d) shows the distinctive difference between the average foreground and background ranking score for all colour and grayscale images respectively.	129
4.8	Algorithm sequence for border seeds combined with the foreground estimate. The input image I in 4.8(a) is passed into SR in 4.8(b), where the SR moment centroid is used to centre the Gaussian centredness map S_G in 4.8(c). A high-frequency response S_L is computed by the 3×3 Laplacian and the $K_B \times K_B$ box filter in 4.8(d). The final foreground estimate based on image frequency is S_{HSF} in 4.8(e) which can be compared with the ground truth masks in 4.8(f). S_{HSF} is threshold using the OTSU method in 4.8(g), and then combined with the border nodes S_B from 4.8(h), for a better estimation of the background border seeds in 4.8(i), where S_i are the replacement background nodes. S_B can be compared with the ground truth border seeds in 4.8(j).	130

4.9	WHO parameter sensitivity study. Figure 4.9(a) provides timing for <i>Lab</i> colour-space-only and various $K \times K$ patch size and histogram bin delta angles. Error bars represent one standard deviation of the measured timing. Figure 4.9(b) provides ratio of the WHO parameter variation over the <i>Lab</i> colour-space-only metrics. The metrics are Mean F -measure, Max F -measure, AUC, and MAE for colour and grayscale images. Each data point is the average of all 32,536 images.	132
4.10	Image feature sensitivity study. Figure 4.10(a) provides timing comparisons for various image feature in addition to the <i>Lab</i> colour space. Figure 4.10(b) provides ratio of the various image features over the <i>Lab</i> performance. The same metrics and number of images are used as per descriptions in Fig. 4.9. The image feature cases are AKAZE [10], FREAK [11], BRISK [12], BRIEF [13], SURF [14], and SIFT [15]	134
4.11	Standard dataset salient image comparisons.	143
4.12	SatSeg dataset salient image comparisons.	144
4.13	Saliency detection results for colour images. Column 1: ECSSD, DUT-OMRON, MSRA10K shares the same legend in 4.13(g). Column 2: all figures shares the same legend in 4.13(k).	145
4.14	Saliency detection results for grayscale images. Column 1: ECSSD, DUT-OMRON, MSRA10K shares the same legend in 4.14(g). Column 2: all figures shares the same legend in 4.14(k).	146

4.15	Saliency model timing comparisons. Vertical red line indicates the required real-time performance. The \times markers indicate runs computed using the 64L platform with C++; the \circ markers indicate runs calculated using the 64W platform with windows compiled executables, the \square markers indicate runs computed using 32W platform with MATLAB, the \diamond markers indicate our models ran on the 64L platform with C++. Refer to Table 4.6 for method codes. Refer to Table 4.7 for platform codes.	148
4.16	ISS infrared image threshold level variation, values under each figure represent the intensity threshold value.	149
4.17	FED vs. DoG comparisons. \bar{S}_{fed} is the mean value of $\mathcal{FED}(\bar{\mathbf{I}}, \mathbf{S}_e)$. The FED results are slightly more accurate by preserving the end-effector on the SSRMS.	151
4.18	Pointing phase identification and foreground mask generation. The foreground and background histograms are red and blue lines in Fig. (l) respectively. Refer to Algorithm 4 for sequence details. . . .	153
4.19	Radarsat 3D model test image.	153
4.20	ISS infrared image comparison of contrast thresholding and region detection.	154
4.21	RSM infrared image pre-processing and level-set based pose estimation.	160
4.22	Saliency map comparison of selected images from the grayscale SatSeg dataset. The ‘Ours’ label represent the <i>fst+</i> method.	161
4.23	Performance comparison plots for image saliency using the grayscale SatSeg dataset. The ‘Ours’ label represent the <i>fst+</i> method.	163
4.24	ISS saliency method comparisons. For the ISS infrared video test, Algorithm 6 provides the best foreground extraction out of all techniques.	164
4.25	RSM infrared image pose initialisation.	167
4.26	Monocular infrared image pose estimation.	168

5.1	Envisat pitch-axis rotation images.	176
5.2	Pitch axis pose estimation vs. ground truth.	178
5.3	Testing metric resulting from M number of training images.	179
5.4	Computation time resulting from K components.	180
5.5	Inference metric resulting from K components.	180
5.6	3D environment and coordinate system for training image generation.	181
5.7	Reduction in image resolution using pyramids	182
5.8	Estimated position from different training resolutions.	182
5.9	Estimated orientation from different training resolutions.	183
5.10	Eigenspace error and test search time.	183
5.11	Synthetic Radarsat testing images. Top: corrupted testing images. Bottom: e PCA optimised output.	190
5.12	SpaceX Dragon vehicle training images.	191
5.13	SpaceX Dragon vehicle e PCA optimised training images.	191
5.14	SpaceX Dragon vehicle e PCA optimised testing images. Top: corrupted testing images. Bottom: e PCA optimised output.	192
6.1	Point-based pose estimation pipeline overview.	195
6.2	Elemental building shapes.	196
6.3	Internal model adjacent line point inflation	196
6.4	Camera coordinate system definitions.	197
6.5	Envisat feature matching, top image is SIFT keypoints and matching, the bottom image is Shi-Tomasi corner keypoints and matching.	200
6.6	SIFT vs. BRIEF matches, where $mtch$ is estimated matches, $gdmt$ is good matches, B is BRIEF, and S is SIFT.	201
6.7	Match quality relative to match ratio, B is BRIEF, S is SIFT.	202
6.8	SIFT feature matching outliers.	203
6.9	30-point cylinder SoftPOSIT pose estimation; the red circles are image points, the blue points are final estimated model points.	206

6.10	Number of successes as a function of error tolerances.	216
6.11	Percent iteration time as a function of the error tolerance.	217
6.12	SoftPOSIT pose estimation of the RSM captured by an ICI IR camera.	217
6.13	Qualitative pose estimation results.	224
6.14	Quantitative pose estimation results.	225
6.15	Various estimated Envisat pose, the minimum angle is 0 deg, the maximum angle is 135 deg.	226
6.16	<i>ePnP</i> vs. SoftPOSIT performances; red lines represent run time, blue lines represent position error, magenta lines represent orienta- tion error.	227
6.17	Correspondence points vs. image noise for various GN iterations. . .	227
7.1	Spacecraft image contour, camera coordinate system definition, and the level-set function.	231
7.2	ISS pose estimation level-set functions.	232
7.3	Centre initialisation of the target object.	236
7.4	The enhanced gradient descent results for Envisat (rows 1-2), RSM (rows 3-4), and ISS (rows 5-6) synthetic image pose estimation. . .	239
7.5	Gradient descent results for a single image. For large initial offset between the model projection and the image, the gradient descent is performed by a coarse depth and lateral movement, followed by rotation (as shown by the first row), then by the same combination in fine step adjustments (as shown by the second row). Finally, even smaller step adjustments in all directions are applied simultaneously (as shown by the third row).	240
7.6	Image sequence pose estimation uses the foreground pixel probabil- ity posterior generated from the previous frame. Leakage of back- ground contour is amplified after three frames as shown by the first row. The segmentation mask is improved by using the proposed thresholding and fill technique as shown by the second row.	241

7.7	The posterior of later frames is degraded over time if only using the <i>a priori</i> mask in generating the training histograms (shown by the first row). Using immediate previous frames and Otsu thresholding results in a stronger foreground posterior (shown by the second row). The additional filling of the segmentation mask improves the posterior results (shown by the third row).	242
7.8	Pose estimation results for multiple image frames from an approach maneuver. 3D projection matches the segmented target image with some error in the out-of-plane rotation that is difficult to infer by silhouette registration.	243
7.9	Pose estimation results for multiple image frames from an Envisat rotation (https://youtu.be/8Km--F0mC8E).	244
7.10	Pose estimation results for multiple image frames from a RSM rotation (https://youtu.be/IEMpdNHJwic).	244
7.11	Pose estimation percentage error. Left: Envisat trial. Right: RSM trial.	245
7.12	Envisat pose estimation using synthetic monocular camera image.	247
7.13	Envisat pose estimation using synthetic monocular camera images. Red lines are the computed pose, blue lines are the GT.	248
7.14	ISS CAD model synthetic image pose estimation.	250
7.15	STS-135 ISS docking and undocking phase pose estimation using the region-based method.	251
7.16	STS-135 ISS docking sequence thermal image pose estimation and 3D model pose reconstruction.	253
7.17	STS-135 ISS undocking and proximity flyby sequence thermal image pose estimation. Sequence frames 1, 74, 147, 220, 293, 366, 439, 512, 585, 658, 731, and 793 are displayed respectively.	256
7.18	STS-135 ISS undocking and proximity flyby sequence 3D reconstruction.	257

7.19	STS-135 ISS undocking and proximity flyby sequence pose estimation.	258
1	Coordinate system definitions. \mathcal{F}_{00} denotes J2000-ECI, \mathcal{F}_{co} denotes the location of the CS orbit and locates the CS body coordinate \mathcal{F}_{cb} , it is also the ProxOps local LVLH. \mathcal{F}_{so} denotes the SS orbit and locates the SS body coordinate \mathcal{F}_{sb}	297
2	Example of orbit simulation validation expressed in the LVLH. In subfigure (b), the CW general solution does not make a course adjustment at $Z = -20m$ where the integrated solution takes into account the delta-V adjustment in the SS trajectory.	299
3	ProxOps local coordinate systems. Let the servicing CubeSat's camera coordinate to be the same as its body frame. LVLH is the client CubeSat's orbital frame \mathcal{F}_{co} \mathcal{F}_{00} is the J2000-ECI inertial coordinate system. The attitude kinematic-driver for the SS force its Z -axis to point in the direction of the LVLH.	301

List of Acronyms

1D	One-dimensional
2D	Two-dimensional
3D	Three-dimensional
3DS	3D Studio Max
6-DOF	Six-degrees-of-freedom
AGAST	Adaptive and Generic corner detection based on Accelerated Segment Test
ADP	ADaPtive background subtraction
AHE	Adaptive Histogram Equalisation
AKAZE	Accelerated KAZE
AOS	Additive Operator Splitting
AP	Actual Positive
ARD	Automated Rendezvous Dock
AST	Accelerated Segment Test
ASTM	American Society for Testing and Materials
ATV	Automated Transfer Vehicle
AUC	Area Under the Curve
AVGS	Advanced Video Guidance Sensor
BING	BInarized Normal Gradients
BF	Brute Force
BN	Batch Normalisation
BO	Bayes Optimal
BoW	Bag of Words
BoVW	Bag of Visual Words
Box-LoG	Box-Laplacian of Gaussian
BRIEF	Binary Robust Independent Elementary Features
BRISK	Binary Robust Invariant Scalable Keypoints
CAD	Computer-Aided Design

CenSurE	Centre Surround Extremas
CDF	Cumulative Distribution Function
CLAHE	Contrast Limited Adaptive Histogram Equalization
COCO	Common Objects in COntext
CoG	Centre Of Geometry
CoM	Centre Of Mass
CP	Correspondence Points
CRCC	Cone-Rod-Cone Connections
CS	Client Satellite
CSD	Contextual Self-Dissimilarity
CNN	Convolutional Neural Network
CPU	Central Processing Unit
CW	Clohessy-Wiltshire
DAISY	Daisy shaped descriptor
DeconvNet	Deconvolution Network
DO	Drop Out
DOF	Degrees of Freedom
DoG	Difference of Gaussian
DPM	Deformable Part Model
<i>ePCA</i>	euler-PCA
<i>ePnP</i>	efficient PnP
ECI	Earth Centric Inertial
EoM	Equation-of-Motion
EP	Estimated Positive
ESA	European Space Agency
ETS	Experimental Test Satellite
FAST	Features from Accelerated Segment Test
FC-HSF	False Coloured HiSafe
FC	Fully Connected
FCN	Fully Convolutional Network

FD	Fisher's Discriminant
FED	Fast Explicit Diffusion
FLANN	Fast Library for Approximate Nearest Neighbours
fm _x	fast maximum precision model
FN	False Negative
FoV	Field of View
fst	speed optimised model
FP	False Positive
FPR	False Positive Rate
FREAK	Fast REtinA Keypoint
GEO	Geostationary Orbit
GFTT	Good Features To Track
GMG	Godbehere-Matsukawa-Goldberg
GMM	Gaussian Mixture Model
GMR	Graph Manifold Ranking
GN	Gauss-Newton
GNC	Guidance, Navigation and Control
GPS	Global Position System
GT	Ground Truth
HE	Histogram Equalization
HER	HE by Region
HFoV	Horizontal Field of View
HiSafe	High-frequency Salient feature
HK	Ho-Kashyap
HOG	Histogram of Oriented Gradients
HT	Hough Transform
IBC	Image Boundary Contrast
IC	Initial Condition
ICP	Iterative Closest Point
ILSVRC	Imagenet Large Scale Visual Recognition Challenge

IoU	Intersect over Union
IR	InfraRed
ISS	International Space Station
ISSACS	ISS Analysis Coordinate System
ISSBCS	ISS Body Coordinate System
J2000-ECI	J2000 Earth Centric Inertial
JAXA	Japan Aerospace Exploration Agency
k NN	k-Nearest Neighbour
KPCA	Kernel PCA
LATCH	Learned Arrangements of Three patCH codes
LBP	Local Binary Patterns
LDB	Local Difference Binary
LEO	Low Earth Orbit
LIDAR	LIght Detection And Ranging
LMedS	Least Median of Squares
LoG	Laplacian of Gaussian
LSC	Linear Spectral Clustering
LR	Leaky ReLU
LRF	Laser Range Finder
LRN	Local Response Normalisation
LUCID	Locally Uniform Comparison Image Descriptor
LVLH	Local Vertical Local Horizontal
M-LDB	Modified-Local Difference Binary
MAE	Mean Absolute Error
m AP	mean Average Precision
MBD	Minimum Barrier Distance
MOG	Mixture Of Gaussian
MoI	Moment of Inertia
MR	Maximum Response
MR8	Maximum Response 8 Filter

MRP	Modified Rodriguez Parameters
MSD	Maximum Self-Dissimilarity
MSER	Maximumally Stable Extremal Regions
NASA	National Aeronautics and Space Administration
NB	Naïve Bayes
NGC	Normalised Grayscale Correlation
NMS	Non-Maxima Suppression
OAM	Optimal Affinity Matrix
OB	Optimal Bayes
OE	Orbital Express
OOS	On-Orbit Servicing
OpenCV	Open-source Computer Vision
ORB	Oriented FAST Rotated BRIEF
P3P	Perspective-3-Point
P4P	Perspective-4-Point
PCA	Principal Component Analysis
PDF	Probability Density Function
P_nP	Perspective- n -Point
POSIT	Pose from Orthography and Scaling with Iterations
PPV	Positive Predicted Value
prec	precision model
ProxOps	Proximity Operations
PXS	Proximity Sensor
PYR	Pitch-Yaw-Roll
QFT	Quaternion Fourier Transform
RANSAC	RANdom SAMple Consensus
RC	Regional Contrast
RCNN	Regions with CNN features
RGB	Red, Green, Blue
ResNet	Residual-Network

ReLU	Rectified Linear Units
RF	Radio Frequency
RK4	Runge-Kutta-4 th order
rkHs	reproducing kernel Hilbert space
ROC	Receiver Operating Characteristics
ROI	Region of Interest
RPN	Regional Proposal Network
RRRC	Rod-Rod-Rod Connections
RSM	Radarsat Model
RSS	Root-Sum-Squared
RT	Real-Time
RVR	RendezVous-laser Radar
RVS	RendezVous Sensor
S	Schmid Filter
SA	Simulated Annealing
SatSeg	Satellite Segmentation image dataSet
sBRIEF	steered-BRIEF
Segnet	Segmentation Network
SEEDS	Superpixels Extracted via Energy-Driven Sampling
SGD	Stochastic Gradient Descent
SfM	Structure from Motion
SIFT	Scale Invariant Feature Transform
SLAM	Simultaneous Localization and Mapping
SLIC	Simple Linear Iterative Clustering
SoftPOSIT	Softassign POSIT
SOP	Scaled Orthographic Projection
SpaceX	Space Exploration Technologies Corporation
SPHERES	Synchronized, Position Hold, Engage, Reorient, Experimental Satellites
SR	Spectral Residual
SRCL	Spacecraft Robotics and Control Laboratory

SS	Servicer Spacecraft
SSO	Space Shuttle Orbiter
SSRMS	Space Station Remote Manipulator System
SURF	Speeded-Up Robust Features
SUSAN	Smallest Univalve Segment Assimilating Nucleus
SVD	Singular Value Decomposition
SVM	Support Vector Machine
SVS	Space Vision System
TN	True Negative
TP	True Positive
TPR	True Positive Rate
uSURF	upright SURF
UNet	U-shape Network
VCP	Virtual Control Points
VERTIGO	Visual Estimation for Relative Tracking and Inspection of Generic Objects
VOC	Visual Object Class
WHO	Weighted Histogram of Orientation
WS	Watershed
<i>y</i> BRIEF	eye-BRIEF
YOLO	You-Only-Look-Once

Chapter 1

Introduction

AUTONOMOUS spacecraft relative Guidance, Navigation and Control (GNC) is widely recognized as an important technology that enables various space missions. Relative GNC takes part in automated rendezvous and docking (ARD) [16], sample return from Mars [17], re-supply [18], and spacecraft formation flying [19]. Most importantly, relative GNC is essential in on-orbit servicing (OOS) [20, 21] which includes constellation maintenance [22], spacecraft inspection [23], orbital debris removal [24], space structure assembly [25], and proximity operations (ProxOps) between manned [26, 27] and unmanned autonomous space vehicles [18, 28, 29].

In addition to traditional large-scale spacecraft missions, an emerging need for relative navigation of nano-satellite CubeSat is also taking shape. Innovations and cost reductions in digital and electronic devices over the past 30 years have enabled universities and institutions with budget constraints to gain access to space missions using CubeSats [30]. As CubeSat technology matures, its mission capabilities continue to grow, and their structure designs continue to standardize [31]. Early demonstrations of CubeSat relative formation flying [32] and recent interest in spacecraft and space debris [33] ProxOps [34] have pushed relative navigation requirements to the forefront of CubeSat sensors [35] and GNC software.

1.1 Background

The relative *pose* is defined as the position and orientation of the target vehicle relative to the servicing vehicle. The main goal for the ARD navigation system is to determine the target vehicle pose. This section provides a historical account of the ARD sensing technologies and operation techniques, and this section describes the rationale for selecting camera systems for the GNC sensor, the requirement differences, and the pros and cons of working with cooperative and uncooperative targets.

1.1.1 ProxOps History and Technology

Early spacecraft ProxOps dates back to the 1960's US Gemini programs, where the docking adapter had multiple radar transponders to provide cooperative feedback and using strong flashing lights as the manual optical target [36]. The rendezvous radars were accurate up to 60 to 15 meters and were used in the Apollo and Space Shuttle programs for rendezvous and docking. For the Russian Mir space station, Doppler shift in Radio Frequency (RF) beacons from three sources were used for relative pose estimation. The relative attitude can be calculated using RF beacons within 20 meters from the docking antenna [36]. Automated rendezvous and docking technologies were also independently developed by Europe and Japan for International Space Station (ISS) resupply.

Japan Aerospace Exploration Agency's (JAXA) Experimental Test Satellite (ETS)-VII program completed the first ARD on July 7, 1998, where the chaser Hikoboshi was successfully docked with the target Orihime [37]. In the ETS-VII mission, both the chaser and target spacecraft were equipped with a Global Position System (GPS). Relative GPS positions were used up to 500 meters with accuracies of 0.6 meters lateral and 2.7 meters in range. A rendezvous laser radar (RVR) was used from 500 to 2 meters. The RVR has counterparts on the client to provide passive reflective aid to the primary sensor. The RVR has a range accuracy of 1.2 meters lateral and 0.5 meters in range at 500 meters distance in the worse case. Along with the GPS and RVR, a camera proximity sensor (PXS) targeting visual markers on the client was used from 2 meters to full docking. The PXS allowed for 43 centimetre depth accuracy and sub-millimetre lateral accuracy, 0.3 deg roll error and approximately 0.1 deg wobble error [37].

Since ETS-VII, ARD was conducted by the Orbital Express (OE) and the ISS Automated Transfer Vehicle (ATV). The Defense Advanced Research Projects Agency OE mission also performed ProxOps and ARD between two cooperative spacecraft [38, 39]. Orbital Express used National Aeronautics and Space Administration (NASA) Advanced Video Guidance Sensor (AVGS) comprised of an optical imager and use laser diodes of 808 nm and 845 nm to be reflected from the target spacecraft retro reflectors. By subtracting the two wavelength images, only the target marking is revealed [38]. The AVGS can resolve targets at distance several kilometres away [40]. The OE AVGS has been used to 150 meters with 6 mm error in lateral accuracy and less than 0.2 degrees error in relative

attitude accuracy [41].

Soon after the OE mission, the ATV completed its first ARD manoeuvre with the ISS [42]. The ATV and ISS used GPS receivers for far range relative positioning. In close range, the ATV used the so-called RendezVous Sensor (RVS), video meters that bounced pulsed laser beams off of passive retro-reflectors, and used image pattern analysis to estimate its relative pose. During the final approach phase 250 meters from the ISS, a TV camera on the ISS Service Module provides narrow or wide images of the ATV visual targets to be used by ISS crew for visual monitoring. This monitoring system is based on a dedicated image processing algorithm applied to the TV images to determine the relative pose between both vehicles. The ATV pose estimation is specified to be less than 11.3 meters in the travel direction and 5.7 meters in the lateral direction, this is reduced to 1.5 meters and 0.8 meters when the relative distance between the ISS and ATV is 100 meters. The JAXA H-II Transfer Vehicle [18] also adopts the same RVS technique using a Laser Range Finder (LRF).

Robotic operations carried out by the Space Shuttle's Remote Manipulator System (SS-RMS) used visual markers called the Space Vision System (SVS). The SVS is a system of black circular dots embedded in white circular background viewed by a Close Caption Television. The algorithm distinguishes the dot pattern from the surrounding pixels according to a preset threshold. The most likely target pose can be computed from the combined dot patterns using least squares [43, 44].

1.1.2 Pose Estimation

In typical spacecraft GNC, the primary goal is to control the servicing vehicle orbit location and attitude pointing, this is particularly important when it comes to ARD where spacecraft actuation-commands are computed using the relative pose states. Spacecraft pose estimation is used for trajectory guidance, collision avoidance, target inspection and docking state estimation. The relative pose between the chaser and the target can be used to compute the approach trajectory, recognise the component of interest locations, and to determine target motion, mass, and Moment of Inertia (MoI) properties. The docking manoeuvre is classified as sensor berthing, vehicle connection, robotic grasping, and third-party berthing observations.

To compute the target pose without any prior knowledge is called *pose determination*. By incrementally estimating the target pose through changes in the sensor image is referred to as *visual odometry* [45–47] or *pose refinement*. The pose estimation problem can be classified into *known* or *unknown* spacecraft shape and mass properties. The target vehicle is *cooperative* if measurement aid hardware such as fiducial markers, reflectors, or signal transmitters are installed. The aid hardware allows the target vehicle to be easily distinguished from the surroundings and allow orientation measurements to be unambiguous. A target spacecraft without any aid hardware is considered to be *uncooperative*.

Cooperative

Square or circular shapes of different sizes are typically used in a fiducial marker to indicate orientation direction [35]. Examples of the square patterns are provided in Bošnjak *et al.* [48], where a Moiré pattern [49] and infrared (IR) LEDs are used as target features. The glyph Recognition library is used for glyph extraction.^a Alternatively, an elliptical projection of a circular shape is a well-known geometry problem, the equations for circular projection pose estimation are provided in Shiu and Ahmad [50], Guru [51], Zheng *et al.* [52], and Lu *et al.* [53]. Ogilvie *et al.* [54] used four circular target patterns in the OE Demonstration Manipulator System Probe Fixture Assembly for autonomous visual target pose estimation. Fitzgibbon *et al.* [55] offer methods for least square fitting of points to an ellipse. Tanaka and Sumi [56], Fang *et al.* [57], and Tweddle and Saenz-Otero [58, 59] all provided different designs for circular pattern targets. Cooperative systems are more precise than uncooperative methods; however, the target hardware requires physical modifications of the client vehicle; the cooperative targets have a limited range of operation; and the markers must be visible by the monitoring device.

Uncooperative

Hardware such as fiducial markers, reflective aids, and beacons are not available on uncooperative vehicles. For example, the majority of the Geostationary (GEO) and Low Earth Orbit (LEO) satellites are not physically designed for OOS, ProxOps and ARD. The uncontrolled GEO satellite can have a rotational period as low as 16 seconds or as high as 1850

^a<http://www.aforgenet.com/projects/gratf/>

seconds resulting from Yarkovsky-O'Keefe-Radzievskii-Paddak solar pressure torques [60–62]. More importantly, natural and artificial space debris is generally uncooperative. The European Space Agency's (ESA) Envisat LEO spacecraft is often used as an example of an uncooperative target when it lost contact in 2012. Due to Envisat's significant size and orbit location in the highly populated sun-synchronous orbit of 796 km with an inclination of 98.5 degrees, and is considered to be a high-risk space debris, as a collision with Envisat would result in significant numbers of smaller debris in LEO. Kurcharski *et al.* [63] investigated the spin rate of Envisat to be 1.33 deg/s (a period of 271 seconds) counter clock-wise about an axis that is 62 deg from nadir.

One method for uncooperative target pose estimation is the use of LIght Detection And Ranging (LIDAR) systems, which use laser ranging to generate a three-dimensional (3D) points cloud-map for pose computation. LIDAR are reliable since they are independent of lighting conditions. Several LIDAR units are sold commercially, for example, the MDA Rendezvous Lidar System [64, 65], and the Neptec *TriDAR* shown in Fig. 1.1 [66, 67]. The *TriDAR* has flown on the Space Shuttle STS-128, STS-131 and STS-135 missions to the ISS, it uses a three-dimensional laser sensor and a thermal IR imager [67, 68]. The *TriDAR* algorithm uses an onboard target spacecraft 3D model to compute six- degrees-of-freedom (6-DOF) relative pose in real-time where a variant of the Iterative Closest Point (ICP) technique operates on 100 sparse points from the laser sensor [69–73]. The drawback of the LIDAR is its high power and mass requirements and the range restriction of the LIDAR beam. While the mass and power constraints are tolerable for ISS cargo-resupply spacecraft such as the Space Exploration Technologies Corporation (SpaceX) Dragon and the Orbital Sciences Cygnus, they are incompatible with smaller spacecraft especially in the case of the Cubesat.

Optical images can also be used to estimate the target vehicle pose. Compared to LIDAR systems, photo or thermal cameras have lower power, mass, and space needs. With sufficient lighting, visual images can exceed LIDAR observation range from hundreds of meters to several kilometers. Two parallel cameras can form a stereo system, where target depth can be computed using triangulation [74]. One of the latest space stereo research is performed using the Massachusetts Institute of Technology Space Systems Laboratory's Visual Estimation for Relative Tracking and Inspection of Generic Objects (VERTIGO)



Figure 1.1: *TriDAR* installed on Orbital Sciences Cygnus spacecraft. Courtesy of Neptec Design Group Ltd.

experimental facility [75]. VERTIGO is a continuation of the Naval Research Laboratory's Low Impact Inspection Vehicle program [23, 76]. The VERTIGO stereo cameras are installed on the Synchronized, Position Hold, Engage, Reorient, Experimental Satellites (SPHERES) experimental facility on the ISS. Using Simultaneous Localisation and Mapping (SLAM), the SPHERES modules equipped with the VERTIGO cameras performed fly-around maneuver and built a 3D model of an uncooperative object while estimating the relative pose and velocities. In 2013 and 2014, the VERTIGO SLAM GNC technique was validated using the SPHERES free-flyer nanosatellites maneuvering in a 6-DOF microgravity environment inside the ISS Japanese Experiment Module. Although the VERTIGO-SPHERES experiment achieved its research objectives, the SLAM algorithm was not performed in real-time due to computational constraints. Indeed, the VERTIGO-SLAM algorithm used approximately ten minutes to obtain a solution on a $1.2 \text{ GHz} \times 86$ embedded computer [20, 77, 78]. In general, stereo camera performance is partly reliant on its baseline geometry and therefore restricted by the host vehicle physical envelope. Using multiple simultaneous images from different locations can suffer from platform flexibility resulting from spacecraft thruster firings. Compared to monocular images, stereo imagery doubles the amount of data processing, transfer, and storage requirements. Monocular cameras have the least external hardware requirements; however, it is affected by the same

disadvantages as all camera systems such as sensor resolution, lens distortions, motion blur, sensitivity to extreme space lighting environments, and sensor noise. To this end, the algorithm must compensate for the hardware and environmental challenges; and for monocular camera systems, the algorithm also needs to estimate the depth in replacement of multi-sensor triangulation. The main requirements for the image pose estimation are speed, robustness, and precision.

1.2 Motivation

While fiducial markers have high measurement precision, it requires physical modifications to the target vehicle which in most artificial satellites is not feasible. LIDAR and cameras can be used for uncooperative ProxOps navigation sensing, where cameras are usually smaller, lighter, and less power intensive than LIDARS. For small spacecraft such as the CubeSat, cameras are more desirable due to less constraining requirements. Example of this is the inspection and docking CubeSat from the Surrey Training Research and Nanosatellite Demonstrator program [79]. Image pose estimation may be categorised into three phases. First, the image is pre-processed by performing operations such as pixel intensities [80], noise removal [81], data compression [82], feature identification and enhancement [83], salient feature extraction [84], foreground and background separation [85], recognition [86], target localisation [87] and image segmentation [88]. In the second phase, the target 3D pose is estimated by either registering keypoints, lines, circles, regions, chains or free-form objects with internal models [89, 90] or by tracking changes of the initial image features through time [91]. The 2D image and 3D internal model projection matching are typically computed using iterative algorithms that minimises some global cost function. This process is called the *model-to-image registration problem* or the *simultaneous pose and correspondence problem* [92]. Finally, the estimated pose state may be further refined using stochastic optimisation by combining measurements with dynamic prediction or by statistic consensus for solution robustness; typical methods include Kalman Filters [93] and RANdom SAmple Consensus (RANSAC) [94] respectively. Example of this approach can be found in Sim *et al.* [95], Aghili and Su [96], and Assa and Janabi-Sharifi [97].

Image features, edges, and regions can be distorted by lighting or camera hardware.

Often, cluttered background adds uncertainty to the registration process and adversely affect the predicted pose precision. While statistical methods [94] or templates [98] may reduce uncertainty, there is no guarantee in fully removing the adverse influence of the background. A direct solution is to remove the background [6] from the image by using a trained classifier [88] or through image saliency [99]. The processed image can then be used in the 2D to 3D pose matching [100, 101].

While the specific design reference mission cases investigated by this study may be too coarse for docking alignment, our approach is also useful when the relative distance is too large for LIDAR or stereo algorithms to extract sparse 3D points reliably [102]. Passive cameras can also be used to make coarse pose estimations, confirm precision sensor readings and can be used as secondary navigation sensor backups. Our design reference missions consider the target spacecraft to be at a distance where the entire vehicle is visible in the camera image. This scenario is generally during far-to-mid-range ProxOps or target inspection depending on the target geometry and the camera Field-of-View (FoV).

1.3 Problem Statement

This thesis describes the design of uncooperative real-time pose estimation algorithm focused on providing front-end image processing, foreground extraction, and 2D to 3D pose estimation in the range of 500 to 50 meters, using a monocular monochromatic thermal IR camera sensor.

1.4 Literature Review

Lead by the conclusions of the studied topics discussed and summarised in Sec. 1.5 and Sec. 1.7 respectively, this thesis examines and develops new techniques in five branches of studies in machine vision and deep learning. The categories are target vehicle recognition and localisation, target vehicle foreground extraction, appearance-based, point-based, and region-based pose estimation. Recognition and localisation refers to recognising and identifying the target location using an object bounding box. Target vehicle foreground extraction relates to the pixel-wise segmentation of the target from the image background. To identify all items in the image and their pixels is called *semantic segmentation*. To

extract only the target and disregard the rest of the image, one can use techniques in *background subtraction* or *saliency generation*. Point-based and region-based pose estimation uses 2D-3D model projections of matching keypoint features or target projection silhouette respectively. Both methods belong to the *supervised model-to-image registration* problem, however, the former relies on feature keypoints whereas the latter relies on image segmentation which are two separate branches of study in computer vision. The following sections explore the related work in each field of study respectively.

1.4.1 Target Recognition and Localisation

Target recognition and localisation are determined using machine learning and deep learning techniques; we investigate the use of Bag of Visual Words (BoVW) and Convolutional Neural Networks (CNN) methods to achieve this task.

Bag of Visual Words

Traditionally, object recognition can be performed by classification of image parts using probabilistic models [103], or by edge based features [104], by shape models [105], and the use of textures [106], where texture filters are applied to the image to extract invariant features to train an object classifier. Recognition methods involve using Haar wavelets [107, 108] or Scale-Invariant Feature Transform (SIFT) [15] keypoints to construct BoVW [109, 110] by using a codebook of clustered edge features [104]. Early work in texture discrimination by Malik and Perona [111] models human preattentive texture perception; it convolves a textural image with a filter bank to mimic the function of simple cells in the primary visual cortex that respond to edge orientations. Julesz [112] first introduced the term *texton*, it is intended as a putative unit of preattentive human texture perception [113]. Once obtained from training, these *textons* can form the basis vocabulary to build classification functions in identifying the object of interest. Cula and Dana [114], and Varma and Zisserman [115], both have investigated the statistical categorisation of textons for texture classification. Varma and Zisserman used various filter banks such as the Leung and Malik set [116], the Schmid (S) set [117], and the Maximum Response (MR8) set. This thesis builds on Varma and Zisserman's work by combining the S and MR8 filter banks for texton generation. Parallel to Varma and Zisserman's work [115], Fei-Fei and

Perona [118] investigated the use of local features in generating a codebook for Bayesian learning, or known as BoVW. Bag-of-visual-words is analogous to the text-based Bag-of-Words data representation; it is adapted for feature-based image recognition. This thesis combines both techniques and using SIFT [15] keypoints to gain the robustness of clustered texton and the flexibility of using a codebook dictionary. The details of using BoVW for bounding box generation is provided in Chapter 3 Sec. 3.1.

Convolutional Neural Networks

Another popular method for object recognition and localisation is by forming Histogram of Oriented Gradients (HOG) [119] image features into Deformable Part Models (DPM) [120]. The DPM then use a so-called *latent* Support Vector Machine (SVM) [121] to classify the modelled object. Zhang and Jiang [122] also proposed using HOG based kernel regression for spacecraft detection. The HOG based DPM has been likened to a handcrafted version of the Convolutional Neural Network (CNN) or ConvNet [123], where ConvNets are the machine-learned kernel filters that produce response maps which retain principal image structures. ConvNet has experienced a breakthrough since 2012, and many recent techniques use ConvNets for supervised object classification [124]. In 2012, Krizhevsky *et al.* [86] demonstrated ConvNet's effectiveness on the Imagenet Large Scale Visual Recognition Challenge (ILSVRC)-2012, a large-scale database with millions of images [125]. Zeiler and Fergus [126] showed kernel weights learned on large image datasets outperform handcrafted ones and can be transferred to smaller datasets [127]. With the development of deeper [2, 128, 129], and wider [3, 4, 130] networks, ConvNets have become the engine of choice for image classification and object detection [87, 131–135].

Recent advances in ConvNet models have improved accuracy in the classification of image objects [4, 87]. The You-Only-Look-Once (YOLO) network [135] and Single-Shot-Detection [133] with MobileNet network [136] are designed for fast object detection on small resource-constrained platforms. Huang *et al.* [137] recently produced a comprehensive review of the newest object detection networks, where he showed higher accuracy using *Inception-ResNet* or *ResNet-101* for classification and using *Faster-RCNN* for bounding box detection. The *Residual-Network* (ResNet) [2] increases the network depth to 152-residual layers in comparison to the maximum 8-layers AlexNet [86], 19-layers VGG

network [128] and 22-layers GoogLeNet [3]. Instead of only increasing network layers in the depth direction, a wider network offers higher performances. *GoogLeNet* introduces the *inception modules* that uses kernels of various size perception apertures. Szegedy *et al.* [4] combined the ResNet with the Google Inception module [3] to form the *Inception-ResNet*. Integrating the state-of-the-art ConvNet classifiers *ResNet* [2] and *Inception-ResNet* [4] with the object detection engine *Faster-Regions with CNN features* (Faster-RCNN) [87] result in a network can detect the target location with high precision. Chapter 3 Sec. 3.2 provides details in comparing the ResNet and Inception-ResNet methods using monocular camera images for target spacecraft detection and localisation. This thesis also provides a simulation environment to produce realistic orbit and attitude motion of the target and chaser spacecraft.

1.4.2 Target Extraction

Bounding boxes can crop the target object, but background bodies may still exist within the box region. Precise segmentation allows extraction of the target by predicting the entire surrounding background or by extracting the salient image regions. Recent developments in background subtraction, ConvNets and image saliency detection could produce pixel-wise target extractions. This section provides details of background subtraction, ConvNets, and image saliency.

Background Subtraction

Several studies have summarised background subtraction methods [85, 138]. A simple approach is to use a first-order recursive filter by Heikkilä and Silvén [9], hereon forward called ADP, in the form of $B_{k+1} = (1 - \alpha)B_k + \alpha I_k$, where α is an adaptation coefficient, B_k is the adaptive background image and I_k is the incoming frame. The difference frame is processed by applying Otsu's threshold [139]. This adaptive scheme allows distinction of active foreground pixels from inactive background ones. While this simple approach can remove static backgrounds, it falls short when the background is in motion. The OpenCV 3 library provides several other background subtraction techniques, these are the Mixture of Gaussian (MOG) method based on an adaptive Gaussian Mixture Model (GMM) [5] and two enhancements to the GMM method by Zivkovic *et al.* denoted by MOG2 [6]

and KNN [7]. Finally, the Godbehere-Matsukawa-Goldberg (GMG) denotes a method by Godbehere *et al.* [8] combining statistical background image estimation, per-pixel Bayesian segmentation, and an approximate solution to the multi-target tracking problem.

Convolutional Neural Networks

Foreground extraction may also be achieved through image segmentation, which is a core computer vision problem. Early work investigated edge contours [140], colour [141] and texture based features [142] to create separation boundaries. Foreground and background separation can also be performed using optical flow [143], and salient object detection [144]. *Semantic segmentation* assigns labels to each individual pixels to create an annotation map; consequently, ConvNets are well suited for this task [88, 145–150]. This investigation use techniques based on the *Fully Convolutional Network* (FCN) ConvNet family proposed by Long *et al.* [146, 151] Ronneberger *et al.* [147] (UNet), Noh *et al.* [148] (DeconvNet), and Badrinarayanan *et al.* [88] (Segnet). Newer ConvNet segmentation methods such as ENet [152] and Mask-RCNN [153] were developed after our investigation, and can also be evaluated in the future.

This thesis first evaluate recognition networks for image identification, then upgrades the evaluated network to form Segnet-like autoencoders for pixel-wise classification. Simonyan and Zisserman [128] increased the depth of AlexNet [86] with the so-called VGG model and demonstrated a 3×3 receptive field is equivalent to a larger 7×7 kernels when convolutional layers are increased. The smaller kernel reduces the number of parameters in the network allowing more convolutional layers, and the VGG results show higher precision over AlexNet on ILSVRC-2012. Ronneberger *et al.* [147] presents a mirrored decoding ConvNet forming a U-shape Network (UNet) with *skip-layers* to segment cell images. The *skip-layers* concatenates the encoding network layers to the decoding side to preserve input structure [154]. Redmond *et al.* [134] used Darknet [134] in the *You Only Look Once* (YOLO) network for object localisation. Darknet is similar to VGG, but oscillates between convolutional kernels of one and three-dimensions. AlexNet [86], UNet [147], VGG [128], and Darknet [134] are manageable networks selected for this investigation. The recognition networks are modified into auto-encoders with skip-layers in this investigation. Details of our ConvNet-based semantic segmentation is provided in Sec 4.1.1.

Image Saliency

The disadvantage of the ConvNet approach is the need for millions of labelled images to train an uninitialised network [86] or several hundred relatively unique labelled object images for transfer learning [127]. In many instances, similar training images can lead to over-fitting and instability during training. Image saliency is an alternative approach that can be image-driven and does not require tedious training. Image saliency detection is a natural technique primates use to focus resource-limited attention to only the most relevant region or target from an input image [155]. Saliency detection has been one of the main focuses by the vision community for several decades. Some of the visual saliency application includes action recognition [156], scan path prediction [157], interest region proposal [158], image compression [159] and many others. This thesis use image saliency to distinguish the target spacecraft from a cluttered Earth background. We use the so-called bottom-up image-driven approach to detect and extract the spacecraft target in real-time which will reduce the overall pose estimation processing time and prediction errors.

Salient feature detection can be categorised as top-down and bottom-up approaches. The top-down approach often have some scene understanding [160], having memories of the past feature structures, either globally [161, 162] or locally [163–166], while performing supervised learning with class labels. The top-down approach resembles functions in the higher regions of our brain forming more complicated and expensive classification to comprehend the global coupling [167]. Often, the top-down approach is combined with bottom-up techniques for generating the saliency map [168–172]. The bottom-up approach is image driven and pre-attentive, it originates from uniqueness, rarity, irregularity, and surprise. These qualities are closely related to stimulations to our visual system. The non-salient image signals are pre-filtered before the most useful information is used for higher processing [171].

Various ConvNet-based saliency detection models have been recently introduced [162, 173–177]. However, because our application has limited training input, this work do not compare with the network-based approach that requires large amounts of labelled training data. Driven by the unsupervised real-time requirement, we choose the bottom-up approach as the starting point of our saliency generation method. Given the large body of work on image saliency from the vision community, we restrict our discussion of image saliency

mostly to bottom-up techniques. For a more comprehensive list of saliency models, a survey by Borji [155, 178] on 41 state-of-the-art methods and an on-going effort in maintaining comparative saliency techniques can be found on their project website.*

Seminal thoughts on image saliency are traced to Koch [179], where founding ideas of *proximity* and *similarity* lead to the early Itti [84] model which proposed an attention system based on centre-surround, image features, and a so-called the *Winner-Takes-All* neural network. This methodology can be categorised as one of the biologically inspired ways of predicting eye fixation. Other eye fixation predictions include Spectral Residual (SR) [180], Phase Spectrum of Quaternion Fourier Transform (QFT) [181], and spatiotemporal attention [182]. In SR [180], the image is separated into salient *innovation* and non-salient *prior knowledge*, and the *innovation* can be computed using the residuals after removing the log spectrum. Quaternion Fourier Transform extended this idea by using a quaternion made from colour, intensity and motion features between frames [181]. Zhai and Shah proposed the temporal attention model [182] where the Scale Invariant Feature Transform (SIFT) [15] operator is used to find frame-to-frame correspondence and motion contrast in identifying attention to moving objects.

On the other hand, methods based on salient object detection of global or local regions are computed through systematic raster scans [183–185]. Saliency maps may be generated by global frequency filtering [186, 187], geodesic [188], or local centre-surround [189]. Other methods compares local resemblance [190, 191], directly using colour contrast [167, 192] or colour similarity [99, 189]. Non-colour based models were also explored, mainly the use of image texture [166, 193], image features [194] for still images, and optical flow [195] for motion sequences. In an example of the grayscale image saliency detection, Jung *et al.* [196] proposed the use of orientation from image gradients to form a histogram that is later mean shift filtered. The saliency map is computed from the gradient magnitude and the cosine distance between the local directional residual from gradient orientation and the nearest dominate angle. However, using orientation features alone will only produce blurry attention regions with low precision [196].

Early methods in saliency detection are pixel- based [184, 189], while the results can have fine resolutions in defining the foreground boundaries, our experience has been that

*<https://mmcheng.net/salobjbenchmark/>

the pixel-based approach is too computationally intensive for real-time implementation. The recent developments in image superpixels [197–199] has been adopted by many of the modern works in saliency generation [99, 164, 200–206]. In particular, the Simple Linear Iterative Clustering (SLIC) superpixel is the defacto choice for graph manifold based methods [99, 203, 204, 207, 208]. With so many methods in saliency generation, some general trends may be observed: the use of centre-surround either by foreground localisation or from borders for background seeding; the segmentation of local regions and similarity comparisons of these regions; the combination of multiple response maps to enhance the most salient features. For example, a recent publication by Li *et al.* [206] contains many of the typical components of a modern bottom-up saliency model, such as a Gaussian Mixture Model decomposition with hash-based clustering combined with SLIC based centre prior uniqueness to produce the foreground saliency map. Li [206] then combined this foreground map, border pixel seeded background map, and a smoothness difference, in a manifold ranking optimisation scheme for the final saliency generation. Details of image saliency generation and performance are provided in Chapter 4 Sec. 4.2.

1.4.3 Appearance-based Pose Estimation

Two dimensional image pose estimation is a core computer vision problem. The most popular image-based spacecraft pose estimation use either a *non-model-based* or *model-based* approach. Non-model based approach include the use of stereo egomotion estimation [209], optical flow [210], and Structure from Motion (SfM) [211]. Augenstein and Rock [212] used SIFT features in SLAM or recursive SfM for frame to frame tracking of the target spacecraft. The non-model based approach does not require knowledges of the target object but is disruptable by tracking feature loss. Model-based approach require knowledges of the target shape; they include feature-based model tracking [213], template matching [214], contour tracking [215], articulated object tracking [216], and point correspondence [100]. Rosenhahn *et al.* [89, 90], and Lepetit and Fua [217] surveyed various methods using feature-based structures such as points, lines, circles, and chains to free-form objects such as active contours and implicit surfaces.

Space imagery is unique in many cases where the entire spacecraft is in front of a black space background. In these instances, one may directly compare the appearances

of the target with stored images that are transformed and compressed in eigenspace; this method is referred to as Principal Components Analysis (PCA) [82]. The PCA method allows efficient compression of the target image content and comparison; a more extensive discussion on PCA is provided in Chapter 5. The PCA method breaks down however when the target vehicle cannot be fully observed and if the target is passing over the Earth. While segmentation or saliency detection techniques discussed previously may aid in the removal of background clutter, faster and more robust techniques are available that take advantage of image features and region shapes. Miravet *et al.* [21] extract the target spacecraft from its background and scores basic two-dimensional (2D) geometric shapes to the image blob; this approach may be sufficient for narrowly predefined missions but can be easily disrupted by projection geometry, illumination, viewpoint, shadowing and rotational changes. The space image dataset published by Lingenauber *et al.* [218] shows under real-world lighting conditions, the resulting images can be highly complex.

1.4.4 Point-based Pose Estimation

Methods in image registration often involve point or line correspondence. Early work in correspondence alignment often results in geometric, probabilistic, and iterative schemes. For example, Haralik *et al.* [219] use wire frame object position by using cones in transforming a 2D to 3D problem to a 3D to 3D matching problem. Lowe [220] used the probabilistic ranking of invariant structures to search for matches and use spatial correspondence to match 3D models. Pinjo [221] used 3D moment tensors to obtain orientation transformations. Thompson and Mundy [222] defines a vertex-pair to determine the affine transformation between a 3D model and a 2D image using clustering. Gavrila and Groen [223] use *geometric hashing* to identify an object in the scene by indexing invariant features of the model in a hash table. Grimson and Huttenlocher [224] derived matching conditions based on a statistical occupancy model and provided a method for setting threshold, so the probability of a random match is minimized. Jurie [225] used a Gaussian distribution to model image noise to improve the probability of matching, and used a recursive multi-resolution exploration of the pose space for object matching. Keypoints used in the correspondence are usually designed to be stable and reproducible under camera or target state and environment variations. The most popular image features such as SIFT [15], Speeded-Up Robust

Features (SURF) [14], or Accelerated-KAZE (AKAZE) [10] are mostly invariant to scale, rotation, and to some degree affine transformations. Transformation invariance allows for stable point features when transitioning through each frame. The image feature also contains handcrafted descriptor vectors as identifiers for keypoint matching between images [226, 227]. Image features usually do not depend on the target object, alternatively, Zhang *et al.* [228] introduce a cooperative keypoint where Box-Laplacian of Gaussian (Box-LoG) kernel is created based on the known size of the circular fiducial marker. Corners computed from the Harris Matrix [229] or FAST [230] are also used as keypoints instead of blob extrema such as SIFT and SURF. Details on image feature design and performance are provided in Chapter 2 Sec. 2.3.

While edge features have many desirable and invariant properties, they are more difficult to manage than point features from corners or blob image extrema. Two-dimensional point registration is called the perspective- n -point (PnP) problem. Diaz and Abderrahim [231] and Arantes [232] extracted spacecraft point features through Hough Transform (HT) and used an iterative point correspondence based in *Pose from Orthography and Scaling with Iterations* (POSIT). Post *et al.* [227] use image features SIFT [15], SURF [14], and Oriented Features from Accelerated Segment Test (FAST) and Rotated Binary Robust Independent Elementary Features (BRIEF) or known as ORB [233] with Fast Library for Approximate Nearest Neighbours (FLANN) [234] matching and efficient Perspective- n -Point ($ePnP$) [235] as the pose estimator. Chen *et al.* [236] demonstrated 6-DOF pose estimation with a moving camera and target by using a combination of the image feature, $ePnP$ and Kalman filtering. Sharma and D’Amico [237] also use points from edge features as inputs for various PnP solvers. Cho *et al.* [238] used GMM in place of the PnP for point registration, where mixtures of corners and blob extrema are the front-end interest points. To improve robustness, Lourakis [239] applied Least Median of Squares (LMedS) estimator as an alternative to RANSAC [240]. Furthermore, a back-end Kalman filter may better stabilise the tracking process [241–243].

Perspective- n -Point

The PnP approach computes the target body pose relative to the camera by image keypoints and model projection points correspondence. Perspective- n -Point methods can be

classified as *iterative* and *non-iterative*. The iterative approach minimizes an energy function to match the estimated image point to the true image. Iterative methods include $O(N)$ *Direct Linear Transformation* [244] and the $O(NM)$ *SoftPOSIT* [100], where N and M is the number of image and model projection points respectively. The LHM solver by Lu *et al.* [245] is widely accepted as a highly precise iterative P_nP solution. Schweighofer *et al.* [246] propose an $O(N)$ global optimal solution using the SeDuMi solver. Recent iterative P_nP solvers includes works by Fan *et al.* [247], SoftSI by Zhou *et al.* [248], *Scaled Orthographic Projection* by Sun *et al.* [249], and Maximum-Likelihood- P_nP by Urban *et al.* [250].

Close form non-iterative solution includes three-point (P3P) [251, 252] or four-point (P4P) [94] correspondence using geometric or algebraic closed form formulas. Lepetit *et al.* [235, 253] introduced the $O(N)$ close form eP_nP . eP_nP uses four *Virtual Control Points* (VCP) to span any number of the model correspondence points and use inter-VCP geometric distance constraints to solve the VCP in the camera frame. Gao *et al.* [254] and Ferraz *et al.* [255] improve the eP_nP method by adding iteration on the depth matrix and outlier rejection respectively. The Robust- P_nP [256] solution retrieves the roots of a seventh order polynomial resulting from least square minimization of P3P problems. Zheng *et al.* [257, 258] propose two direct minimization methods using Gröbner basis solver resulting in Accurate-and-Scalable- P_nP and Optimal- P_nP . Kneip *et al.* [259] and Nakano [260] both proposes extensions to the *Direct Least Squares* [261] method to include non-central camera rays and by global optimization with Cayley parameterization respectively.

In this work, the *SoftPOSIT* [100] method for frame-to-frame tracking and pose estimation is used. For small pose variations, SoftPOSIT is stable and can be computed quickly. The *SoftPOSIT* method [100] combines iterative POSIT [262] and *Softassign* [263] matching. A global cost function reduces the difference between the projection points with the image points. The correspondence and the pose are simultaneously iterated using deterministic annealing [264] to minimise the projection error cost function. Details of the P_nP point-based pose estimation are provided in Chapter 6.

1.4.5 Region-based Pose Estimation

Keypoints based on image corners [229, 230] and extrema image blobs [10, 15] can either be overly numerous or insufficient under extreme lighting or non-distinctive visual targets. The feature descriptors used for keypoint matching can be time-consuming to compute, and can be erroneous especially under affine or viewpoint transformations and illumination distortions. Edge-based pose estimation [265] represent stable features to track with many invariant properties, but extracting relevant edge features and discarding line clutters need robust decision tools since the edge and feature point image can be cluttered with unstable and unuseful entities such as those shown in Fig. 1.2. On the other hand, the region-based pose estimation uses invariant region boundaries with greater recognition properties.

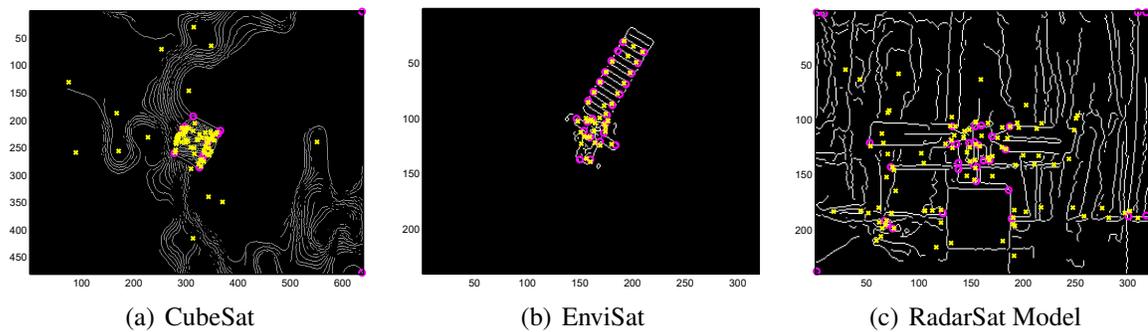


Figure 1.2: Canny edge lines in white, Harris corners in magenta circles, and SIFT keypoints in yellow crosses. When viewed with a non-illuminated space background in the case of the Envisat, the keypoints and edges are invariant inputs for pose estimation and tracking. Line and keypoint features require robust classification when there is a cluttered background.

The region-based pose estimation approach evolved from the image segmentation problem. Image segmentation is a core problem in computer vision; typical solutions includes a mixture of complex top-down classification [88] or simple bottom-up image-driven techniques [139]. Whichever the method, many believe segmentation is too high-level task for purely image-driven methods to succeed [266]. Seminal papers by Kass *et al.* [267], and Mumford and Shah [268] introduced concepts of energy-minimising active contours, the so-called *Snakes*, and the combined minimisation of region smoothness and boundary length respectively. Caselles *et al.* [269] and Chan and Vese [270] followed by adopting an implicit region boundary representation using the level-set function pioneered by Osher

and Sethian [271]. The level-set function provided advantages of implicit boundaries without specific parameterisation and evolving embedding functions that can elegantly describe boundary topology changes such as splitting and merging. Additional contributions by Kervrann, and Heitz [272], Zhu and Yuille [273], and Leventon *et al.* [274] instilled maximum likelihood and maximum a-posteriori criterion to estimate the region boundary with statistical parameterisations. Leventon *et al.* [274], and Rousson and Paragios [275] further stipulated the use of PCA shape template to align the region boundary. Cremers [276] extended the use of kernel PCA [98] to the PCA template and used affine flow fields [277] in discriminating the object from its background. Brox and Weickert [278] and Bibby [279] refined the shape prior, pixel-wise posteriors and proposed a framework for segmenting multiple regions. Rosenhahn *et al.* [280] combined 3D pose estimation with the level-set formulated region segmentation. The two approaches complemented each other to produce a framework for the region-based pose estimation [281, 282]. Additional work by Prisacariu and Reid [101], Hexner and Hagege [283], and Tjaden and Schömer [284] ushers in the modern use of level-set region-based pose estimation for real-time applications.

Some of the newly developed approaches in level-set based segmentation and pose estimation are as follows: Li *et al.* [285] proposed a distance regularised level-set evolution to avoid numerical errors. Dambreville *et al.* [286] projected 3D models on to the 2D image and used region-based active contours for shape matching, his method showed robustness to occlusions. Jayawardena *et al.* [287] use contours generated from 3D models as a projection in performing so-called intra-object segmentation. Prisacariu and Reid used posterior membership probabilities for foreground and background pixel in support of real-time tracking in the PWP3D model [101, 288]. Prisacariu and Reid also used nonlinear minimisation of the image-driven energy function in the learned latent space for segmentation and robustness to occlusions [289, 290]. Gong *et al.* [291] proposed a particle filter-based segmentation with a shape-aware level-set function that incorporates a joint view-identity manifold to model the target shape. Zhao *et al.* [292] used particle filter stochastic optimisation in solving the energy functional. Perez-Yus *et al.* [293] used a combination of depth and colour for robustness to occlusions in the level-set segmentation and pose estimation. Prisacariu *et al.* [294] used a simplified level-set approach to avoid global distance computation for 100 Hz PC operation. Hexner and Hagege [283] proposed local templates to

enhance the PWP3D global framework. Tjaden and Schömer [295] proposed a pixel-wise optimisation strategy based on a Gauß-Newton-like algorithm using linearised twist for pose parameterisation. Tjaden and Schömer [284] also proposed a temporally consistent, local colour histogram as an object descriptor for the level-set pose estimation template. This template consists of four different rotations within the image plane ultimately resulting in 144 base templates forming an icosahedron. Swierczynski *et al.* [296] proposed to merge an active dense displacement field with the level-set formulation. The general trend for level-set region-based pose estimation research is moving in the direction of enhancing the internal model compatibility. We take a different approach to improve the external input image to achieve the same end objectives. Details of the region-based pose estimation is provided in Chapter 7.

1.5 Research Approach

Our goal is to develop a pose estimation technique using monochromatic monocular camera images. The topics of this thesis are selected based on the performance of each investigated method. To generate spacecraft images, we developed a simulator using dynamic orbit and attitude models. The details of the motion simulator are provided in Appendix B. Our primary design reference case is an infrared video sequence of the Space Shuttle Orbiter (SSO) docking and undocking from the ISS captured by the SSO *TriDAR* camera. The ISS video sequences include black space background and Earth passage. We generated image sequences using synthetic 3D models for a CubeSat, Envisat, the ISS, and a reduced scale imitation of the Radarsat called the RadarSat Model (RSM). We also captured laboratory video sequences of a CubeSat testbed and the actual RSM using photo and thermal cameras respectively. We use a kinematic driver to compute the ground truth for the synthetically generated spacecraft videos of Radarsat, Envisat, and ISS 3D models, and use a dynamic orbit simulator for the CubeSat motion. The laboratory RSM does not have precise ground truth pose value and pose comparisons are viewed qualitatively through 3D model projection video overlay. The laboratory CubeSat localisation and the spacecraft foreground ground truth are generated manually. The ISS infrared video ground truth is available in range only measured by the *TriDAR*.

We begin our investigation using corner keypoints and an iterative P_nP method. Two

immediate challenges from using real-world images is a lack of keypoints that belong to the target vehicle and incorrect pose match. For the first challenge, we investigated image processing techniques to improve image illumination, generating keypoints from straight lines, and using various image features. The conclusion of the image feature performance study suggested using corner keypoints, and binary feature descriptors have the best balance for efficiency and precision. Using the binary feature, however, still lack the precision when applied to the laboratory and ISS spacecraft images. We improve on the binary feature by creating the *y*BRIEF descriptor. For the second challenge, we added statistical outlier removal, evaluated non-iterative P_nP solvers, and investigated other pose estimation techniques such as appearance-based and region-based methods. A persistent issue for all pose estimation techniques is the erroneous selection of background features. To this end, we investigate into foreground extraction to directly remove unwanted clutter from the image. First, we examined bounding box generation using machine learning and deep learning techniques. Next, we investigated more precise semantic segmentation using deep learning networks. Experience of bounding box and semantic segmentation all points to the difficulty in obtaining sufficient labeled training images; this led our investigation into unsupervised image saliency methods in foreground extraction. Finally, we combined the most effective image processing method with foreground extraction and region pose estimation.

We limit our application to model-based methods out of practical considerations for future space missions. We also demonstrate our approach can be extended to semi-unknown scenarios as well. As previously mentioned, our investigation experiences lead us to break the pose estimation problem into two phases: an image processing and foreground extraction phase, and the pose estimation phase. In the image processing phase, our goal is to simplify and increase the image feature distinctiveness. We investigate and develop traditional handcrafted image features, performed target recognition and localisation using machine learning and deep learning techniques. We also examined methods of background removal, foreground segmentation using autoencoder ConvNets and image saliency detection. With each investigation, we introduce novel concepts in feature descriptor generation, object localisation, autoencoder networks, and image saliency generation. Our pose estimation development focuses on three main methods: appearance-based, point-based, and

region-based. We applied PCA appearance matching to the spacecraft problem and tested Euler-PCA to demonstrate its resilience to obstruction occlusion. We estimate the target spacecraft pose using iterative and non-iterative PnP methods, namely, SoftPOSIT and $ePnP$ respectively. We also improved point correspondence robustness using RANDOM SAMple Consensus (RANSAC). Our internal model is created using basic element geometries and by performing point inflations along the element edges. The same point inflation is applied to Hough Transform lines extracted from the image. We provide techniques in point occlusion removal to improve correspondence. We enhance the SoftPOSIT precision by introducing a novel initialisation process. We added RANSAC and a novel tracking sentinel to increase robustness. Finally, we investigated region-based pose estimation, where we improve the PWP3D pose estimation method by upgrading the initialisation and gradient descent procedures. The PWP3D approach uses the level-set segmentation technique and pixel statistics. Our improvements include the development of a unique pre-processing false-colored high saliency feature detection which increased the robustness of the gradient descent process.

1.6 Contributions

This thesis developed original techniques, applied new computer vision methods to spacecraft pose estimation, and developed datasets for future research by the vision community. The most significant contribution of our work is the end-to-end process where we developed an automated Earth background detection with novel spacecraft foreground saliency extraction combined with initialisation and gradient descent improved region-based pose estimation. The contributions of this investigation are organised in order of greatest to least significance as target extraction, region-based pose estimation, target recognition and localisation, point-based pose estimation, and simulation environment.

1.6.1 Target Extraction

We applied standard ConvNets in spacecraft recognition and built compact autoencoders for image semantic segmentation. We also developed unsupervised image saliency detection methods. Details of our contributions are itemised in order of greatest to least significance as follows,

1. We provide three enhanced versions of GMR-based image saliency generation optimised in mean F -measure, maximum Area Under the Curve (AUC), and minimum computation time with high performance under 50 ms per frame on average. Our optimised precision model is $3\times$ faster than the original GMR method with lower Mean Absolute Error (MAE) and higher mean F -measure. Our fast maximum precision model is $10\times$ faster than the original GMR method with higher precision versus recall. Our speed optimised version provides a novel seeding method using the latest saliency principals, a Gaussian distributed centredness and high-frequency responses. It is $12\times$ times faster than the original GMR with same or better mean F -measure and MAE [297].
2. We develop our saliency detection for spacecraft navigation but our method can be used for general image applications. We compare our methods with 18 traditional and recent methods in image saliency and discovered state-of-the-art performances. We perform tests of our method on a spacecraft centric dataset and standard datasets ECSSD [298, 299], DUT-OMRON [99], and MSRA10K [167, 169] for colour and grayscale variations totalling 32,536 test images [297].
3. We provide the equation and proof for a PCA based formula to approximate the Optimal Affinity Matrix (OAM) inversion. This technique removes limitations on large matrix inversion, reduces computation time, and stabilises the inversion process [297].
4. We combine a novel weighted gradient orientation histogram distance with the ranking process to increase distinctiveness when colour information is not available. We compared our feature method with conventional image descriptors and found our model has better speed and precision performance. We also compared whitened feature space and L1-distance weight generation against the original L2-distance approach and found L1-distance to be faster while having the same precision [297].
5. We propose a novel spacecraft image background classification scheme based on Difference of Gaussian (DoG) and using pixel statistics to detect Earth passage [300, 301].

6. We introduce an innovative false-coloured High-frequency Salient feature (HiSafe) image to enhance foreground and background pixel histogram distinction, where our approach is proven to be more robust during gradient descent than using unaltered input images [300, 302].
7. We developed a Satellite Segmentation (SatSeg) image dataset complete with target image foreground and background Ground Truth (GT) masks. SatSeg consists of colour, monochromatic and infrared images from real flight data and laboratory models under various lighting and heating scenarios. Our dataset has proven to be challenging for all state-of-the-art methods. We release this dataset freely available to the computer vision community for future developments in salient image generation [297, 300].
8. We tested AlexNet, U-Net, VGG, and DarkNet on a spacecraft image dataset and used selected networks to build SegNet like autoencoders with skip layers. We compare the ConvNet semantic segmentation methods with background various subtraction models [303].

1.6.2 Region-based Pose Estimation

We develop region-based pose estimation using a modified PWP3D method. Contributions from this effort are itemised in order of greatest to least significance as follows,

1. We combine the level-set segmentation approach with novel unsupervised scene recognition and foreground extraction method based on image saliency principles [300, 301].
2. We introduce innovative pose initialisation and gradient descent techniques to speed-up and stabilise the pose convergence process [301].
3. We provide an enhancement to the probability posterior mask using image processing techniques to improve estimation stability [304].

4. We applied the region-based pose estimation process for spacecraft ProxOps rendezvous monocular vision navigation and evaluated the effectiveness of this end-to-end pose estimation model using synthetically generated spacecraft proximity operation images and flight videos from the STS-135 mission [300, 301].
5. We developed an original matrix formulation of the level-set energy partial derivative with respect to the pose parameters while accounting for the pixel-skew factor [304].

1.6.3 Target Recognition and Localisation

We propose a novel method in image processing and apply state-of-the-art bounding box recognition and localisation to spacecraft navigation. Details of our contributions are itemised in order of greatest to least significance as follows,

1. We developed an original biologically inspired image descriptor based on BRIEF called y BRIEF and showed higher performance than the BRIEF descriptor while having comparable performance as the standard image descriptors [305, 306].
2. We applied state-of-the-art ResNet and Inception-ResNet methods to spacecraft localisation in real-time [307].
3. We applied a modified Bag of Visual Words (BoVW) method to the spacecraft localisation problem. Our method uses a texton simplification scheme using PCA and compares five linear, non-linear, and non-parametric classifiers [110].
4. We compare the performance of 12 keypoint detectors and descriptors. We use the top performing image feature for the point correspondence pose estimation.
5. We propose simplified image contrast processing [308] and point inflation on Hough Transform filtered lines for PnP correspondence. The increased number of points allow better alignment compared to only using corner vertices.

1.6.4 Points-based Pose Estimation

We apply the PnP model registration technique for pose estimation; our main contributions are itemised in order of greatest to least significance as follows,

1. We propose a novel homography projection technique in combination with P_nP , RANSAC, and motion sentinel for an end-to-end monocular camera pose estimation solution [309].
2. We develop a generic element geometry scheme to build up spacecraft 3D point model [308].
3. We apply the SoftPOSIT pose estimation method using 3D model corner vertices and improve the SoftPOSIT model using novel initialisation to avoid local minimum trapping [308, 310].
4. We apply the appearance-based PCA pose matching method to spacecraft navigation and combine the adaptive background subtraction to the pose matching scheme [82].
5. We add image culling techniques to the model points [308].
6. We apply the Euler-PCA image occlusion optimisation technique to spacecraft images [82].

1.6.5 Simulation Environment

We developed a dynamic simulation environment to generate realistic orbit and attitude motion. Our simulation contribution is provided in the itemized list in order of significance as follows,

1. We provide dynamic attitude modeling of the target spacecraft using Modified Rodriguez Parameters (MRP) and a kinematic driver pointing formulation for the chaser spacecraft [307].
2. We provide a two body orbit simulation environment with spacecraft thruster model. Our relative motion orbital model is validated using the solution to the Hill's equation [307].

1.7 Thesis Outline

This investigation is organised into eight chapters and supporting appendices. The present chapter introduced the various space missions using spacecraft relative pose estimation and

their methods, the related machine learning and computer vision research, the investigation methodology, and a list of novel contributions. Chapter 2 provides image processing techniques, methods in straight edge filtering, image feature performance comparison, and the *y*BRIEF descriptor. Chapter 3 provides machine learning and deep learning methods for target vehicle recognition and localisation. Chapter 4 provides methods in background subtraction, ConvNet semantic segmentation, and image saliency detection. Chapter 5 provides PCA appearance-based pose estimation and the *e*PCA denoise method. Chapter 6 provides point correspondence methods including SoftPOSIT and *e*PnP and descriptions of using image homography and RANSAC. Chapter 7 provides the application of region-based approach including enhancements to the estimation pipeline. Chapter 8 concludes this investigation and provides future recommendations. Figure 1.3 provides an overview of the thesis organisation.

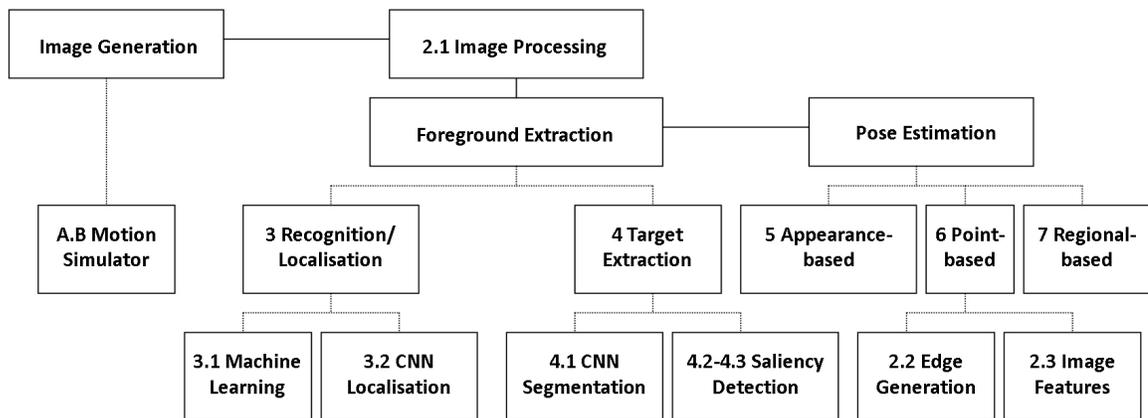


Figure 1.3: Thesis topics organisation chart; solid lines indicate the pose estimation process; dashed lines indicate the investigated method category. Numbers represents section numbers, A.B represent Appendix B.

Chapter 2

Image Processing and Feature Extraction

Images captured in space present different challenges from the terrestrial environment. The sunlight and the Moon or Earth's albedo naturally illuminates a spacecraft. Shadows in space are darker and sharper because there is no atmosphere to absorb or reflect lighting from the surroundings. For the same reason, target spacecraft sunlight reflection appears brighter in space than on Earth. Thermal images may rely on internal spacecraft heating in addition to the external heat absorptions and reflections. Therefore, heated systems and cold regions can result in high illuminations and zero illumination respectively. When the vehicle flies over the Earth, both spacecraft and the background scene can be in motion due to Earth rotation and cloud motion. The adverse effects of space environment lighting are lens flare, bright reflections, hard shadowing, and background clutter. Furthermore, space images are typically in grayscale due to data and hardware restrictions. Hardware wise, imprecise camera construction may cause distortions and defects in the image. For example, distorted camera lens and manufacture tolerance can produce warped images, and the infrared sensor may capture thermal camera internal heating, bias drift, or noise. These effects can cause significant challenges for image analysis and pose estimation precision.

Image processing is needed to enhance the image due to the aforementioned challenges. We also extract distinctive regions of the image for their invariant properties, these image features remain mostly constant with varying camera pose and lighting intensities. We can match these features from frame-to-frame to measure the camera to target relative motion, or we can match these features to internally stored ones to infer the camera pose. This chapter provides techniques in image processing, artificial line recognition, and image feature generation and performance; Chap. 6 will provide methods on how to use image features for pose estimation. Image processing refers to image filtering and image transformations

in order to sharpen Region of Interest (RoI), to smooth image noise, and to enhance image features. Our investigation explores noise reduction, Contrast Limiting Adaptive Histogram Equalisation (CLAHE), and Histogram Equalisation by Region (HER). We explore line generation techniques and propose an original combined line generation method that takes advantage of hard, soft and corner centric line detection. We also provide a procedure for spacecraft detection from artificial lines using Hough Transform (HT) and edge point inflation. The image processing and HT software is provided in Appendix C.2. Feature extraction uses light diffusion and scaling principles to locate interest points or lines, and uses biologically inspired visual description or numerical patterns to generate the most stable and invariant features under rotation, scale, viewpoint, illumination, and image blur changes. We exhaustively compare state-of-the-art image features in Sec. 2.3, and introduces a novel image descriptor called yBRIEF in Sec. 2.3.2 that surpassed the performance of the baseline model.

2.1 Image Processing

We can use image processing to perform noise reduction and contrast enhancement. A common way to reduce noise is by convolving the image with Gaussian or box kernels, where these low-pass filtering operations mix the nearby pixels and smooth the image. However, the smoothing process degrades the edge boundaries and blurs the overall image, which will decrease distinctiveness of line and corner features. An alternative approach is by using a *bilateral filter* [311] to remove noise while preserving edge boundaries. The bilateral filter achieves edge preservation by considering the geometric distance between the neighbourhood centre and the nearby points, and pixel photometric similarity; therefore, far away pixel or pixels from divided regions are ignored. The bilateral filter, however, is computationally expensive and over filtering could lead to the so-called *cartoon effect* where shading inside a boundary is lost; this will cause region blob keypoints and surrounding contrast to lose meaningful feature description. Image normalisation and histogram equalisation can be efficient and effective methods in contrast enhancement. When images are captured in a dark or bright environment, the pixel histogram may only span a small intensity region, and some image features are lost due to lack of contrast. Histogram normalisation spans the intensity histogram to the entire 0 to 255 range where histogram equalisation

(HE) linearises the Cumulative Distribution Function (CDF) of the image histogram over the intensity range; these operations increases the dissimilarity between regions, where edges and local contrast achieves higher distinctiveness than the original image. The drawback of global image normalisation is when the image already spans the intensity range; this often occurs in space images where both extreme lighting from the Sun, camera flare, or for thermal images a heated component, is in contrast to the black space or hard shadows. Histogram equalisation, in some cases, can reduce the information from an image. The linearisation of the CDF assigns the same importance to all image intensities; where details in local regions may be reduced or lost as a result of this indiscriminate linearisation. An example of the normalisation and HE drawbacks is shown in Figs. 2.1(b) and (c) respectively. Sophisticated histogram equalisation technique produces higher performance than the original methods, an example is the CLAHE [80].

Recent advancements to image contrast adjustment are local region stretching [312], particle-swarm optimisation-tuned sectorised equalisation [313], and gradient domain high dynamic range compression [314]. Pizer *et al.* [80,315,316] developed Adaptive Histogram Equalisation (AHE) to enhance biomedical images by applying HE mapping in a region surrounding individual pixels. Zuiderveld later introduced Contrast Limiting AHE [317], to limit contrast amplification and noise. CLAHE uses a so-called *clip limit* to bound the histogram before computing the CDF, which will change the slope of the CDF and the resulting contrast-enhanced image. Some recent improvements to CLAHE are parameter self-determination [318, 319] and alternative region processing [320]. Locas *et al.* [314] showed CLAHE to have superior performance on space images than gradient-domain high dynamic range compression and unsharp mask of log image. A drawback for the CLAHE method is its processing speed; it can be forty times slower than histogram equalisation. We develop a simple alternative to CLAHE, where we apply HE to grid regions called HE by Region (HER). Since the grids are known by design, the sharp difference between grids is smoothed linearly. In Fig. 2.1 we compare an RSM thermal image using normalisation, HE, CLAHE, and HER. Image processing times are 0.11 ms, 0.14 ms, 4.09 ms and 1.49 ms for normalisation^a, HE, CLAHE (8×8), and HER (8×8) respectively. The normalised image is the same as the original image because the image histogram is already spanned over

^aplatform description: AMD FX-83508 – 4.0GHz Ubuntu16

the pixel intensity range. Histogram equalisation improved the image contrast; however, the high heating region in the upper left corner is saturated with high-intensity lighting. CLAHE provides the best-blended result followed by HER, grid regions in the HER image is more defined than the CLAHE image; however, the HER image is nearly three times faster than using CLAHE.

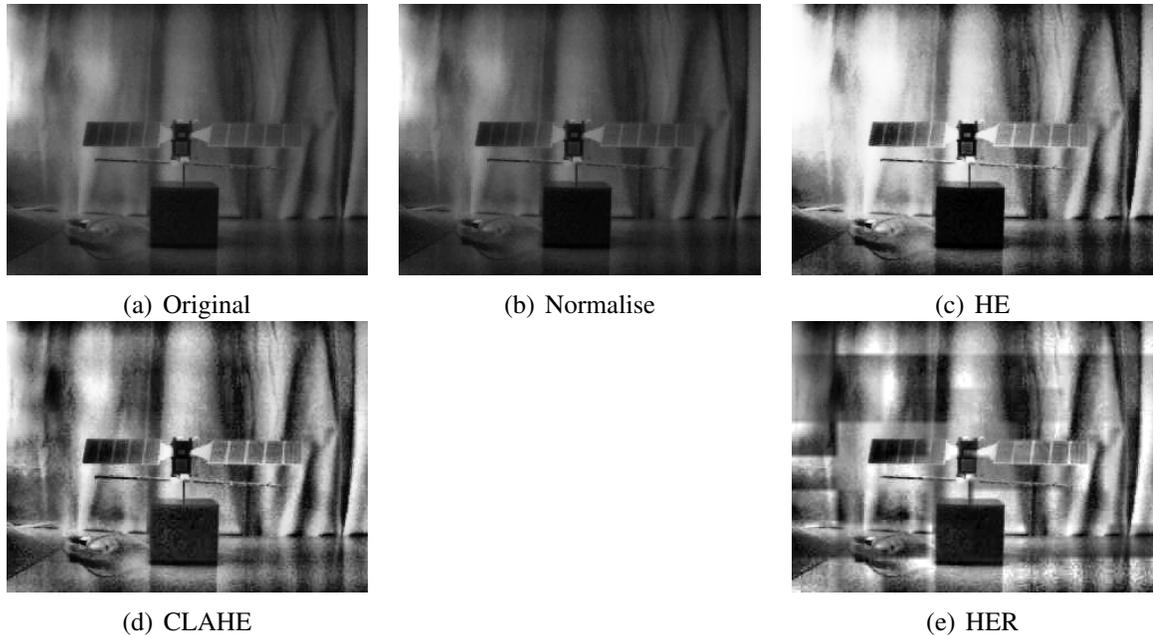


Figure 2.1: Image processing method comparisons. Bands in HER is due to rectangular grids and is smoothed linearly. Since the grid dimensions are known, it can be removed from subsequent edge detection.

2.2 Edge Generation

Boundary edges can be visible under scale, rotation, illumination and viewpoint changes. The invariant properties of edges makes them good features to track; furthermore, we may extract boundary points from the edge lines and use them in point matching algorithms. Edges are strong gradients in the image; they can be computed by convolving the image, *e.g.*, with Sobel [321] or Scharr [322] kernels. We introduce an edge generation algorithm that combines three edge detection methods: Roberts [323], Canny [324], and Harris [229] for hard, soft, and corner-centric edges respectively. Harris and Stephens introduced the landmark paper on the Harris corners [229]; this paper also proposes that edges can be

computed using the structure tensor concept. The Harris edge [229] is different from Sobel [321], Roberts [323] or Canny [324], because it is strongly associated with corners. Figure 2.2 shows the three edge detection methods applied to a synthetic CubeSat image. When each edge detection method are used individually, there can be over or underprediction of image edges. Our tests have shown high performance in generating boundary edges when the three edge methods are combined. The combined response is a weighted sum of

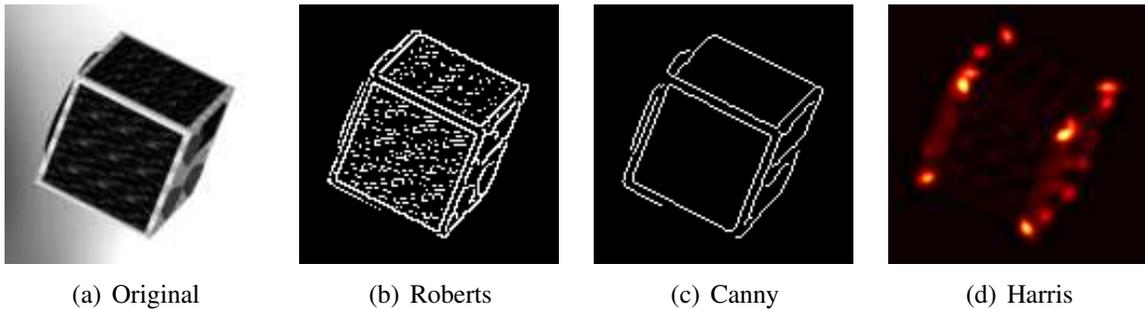


Figure 2.2: Edge detection of a synthetic CubeSat image.

the three edge methods followed by a histogram threshold as follows,

$$I_{edge} = \begin{cases} (aI_{roberts} + bI_{canny} + cI_{harris}) & \text{if } I_{edge} > tol \\ 0 & \text{otherwise,} \end{cases} \quad (2.1)$$

where a , b , and c are constants selected by tests as 0.3, 0.6, and 0.1 respectively. The Canny edge has the highest weighting where Roberts and Harris have fewer contributions than Canny but adds robustness to the edge detection. The histogram tolerance, tol , is selected by tests as 0.5. The edge histogram and response map are shown in Fig. 2.3. The combined edge generation removes the internal edge noise and retains the outer boundaries.

2.2.1 Hough Transform

A naïve edge line may not represent the outer boundaries of the spacecraft; it can be the artefact of the Earth background or spacecraft material textures. A more robust edge is computed using Hough Transform (HT) [83]. Hough Transform converts the edge response

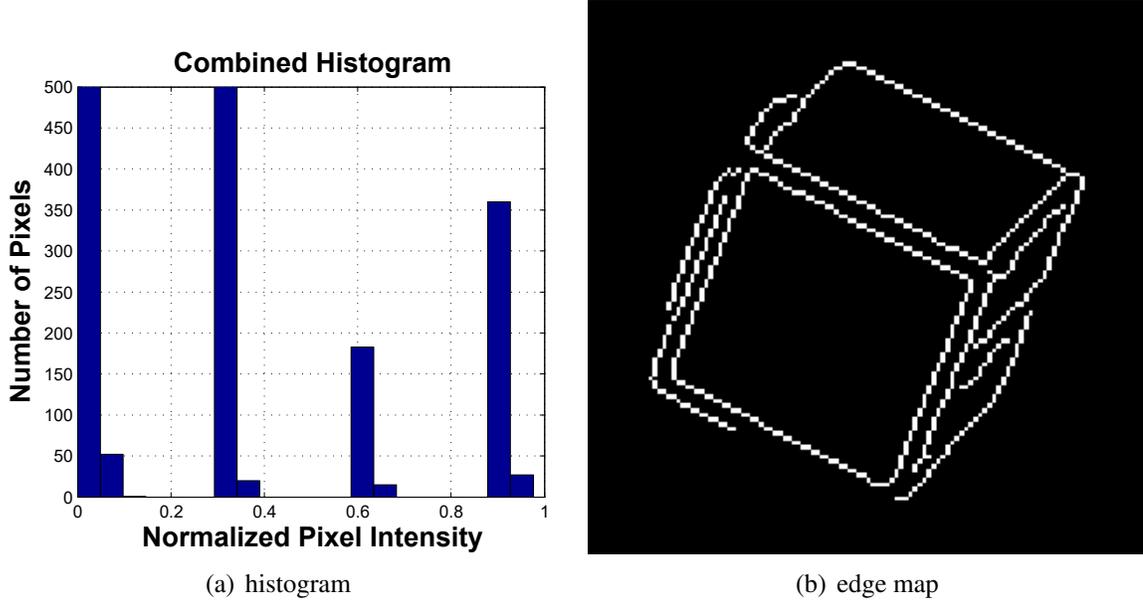


Figure 2.3: Combined edge histogram and response map.

map into a polar space representation through the following transformation,

$$\rho = x \cos(\theta) + y \sin(\theta), \quad (2.2)$$

Where ρ is the Hough space distance, x and y are the image pixel coordinate, and the θ value is varied between -90 to $+90$ degrees. The duality of a point in the image space is a sinusoidal wave in the Hough-space. The intersection of these sinusoidal waves in the Hough-space represents a group of points sharing the same slope and therefore producing a straight line in the image-space. Higher intersection votes in the Hough space represent longer straight lines in the original image. An example of the Hough-space transformation edge is provided in Fig. 2.4. The straight edge produced by the HT operation is depicted in Fig. 2.5. The HT approach resulted in the removal of non-essential lines from the spacecraft edge response map leaving only the outer boundaries of the CubeSat.

Figure 2.6 shows a difficult to process thermal image of the RSM where a hot air heat source in the upper left corner of the image transmit heat towards the RSM by convection. The heated air is dissipated in the middle of the image and does not illuminate the RSM. Figure 2.6 has low contrast around the RSM. The middle image in Fig. 2.6 shows when applying the Canny edge detection directly to the raw input image, only the heat source

can be detected. We implement HER to the raw input image and use the combined edge detection as described in the previous section. We compute connected pixels and remove lines less than 15 connected pixels. Next, we apply HT and Non-Maximum Suppression (NMS) of the HT lines to extract straight lines from the image to remove clutter and non-essential lines. We find the region with the straightest lines and cluster to a centroid point. From this central point, we apply a preset region mask to extract the foreground straight lines; the mask area is an estimated rectangle that covers the likely outer envelope of the RSM. Finally, we overlay the HT lines with the edge response maps shown in the last image on the right of Fig. 2.6. Figure 2.6 demonstrates the effectiveness of our approach in extracting useful line and point features from a difficult to process thermal image, where traditional line detection method fails in the same task.

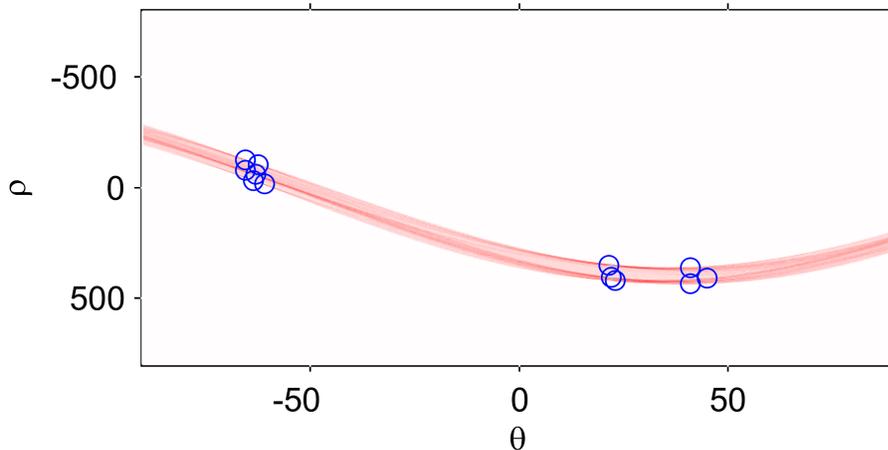


Figure 2.4: Hough-space representation. The top voted points are circled.

2.2.2 Point Inflation

Spacecraft pose estimation and state tracking can be performed using point matching, namely a PnP solver. Robust PnP matching requires sufficient number of image and model points for energy function minimisation. When using the edge based interest point approach, only extracting corner vertices is not enough for robust PnP calculation. To this end, we create edge points by transforming the HT lines computed previously into equal distanced points. We call this line-to-point transformation process *point inflation*. Using the CubeSat example, we linearly inflate 10 points along the HT lines shown in Fig 2.7.

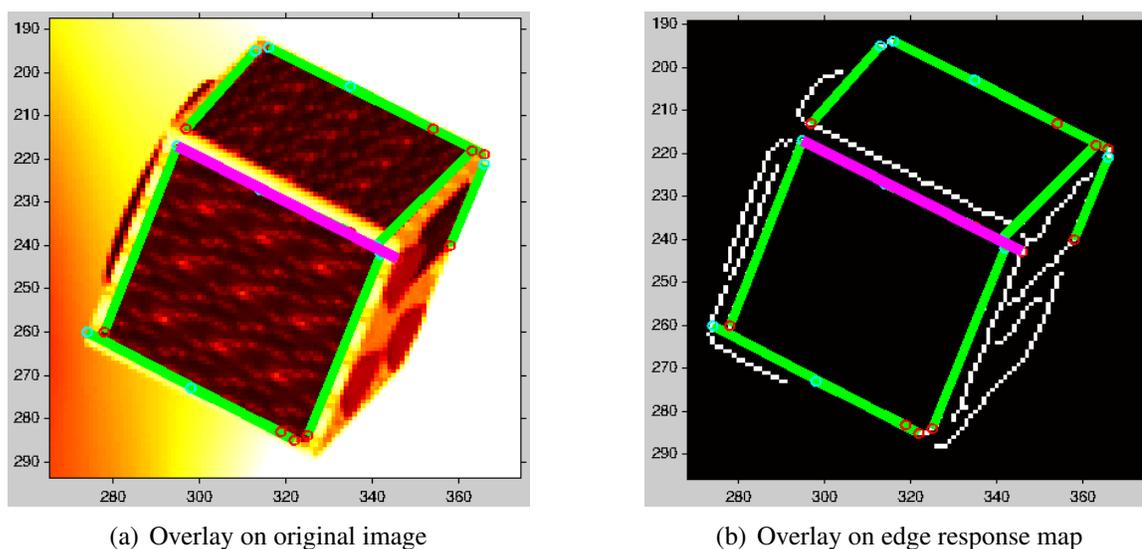


Figure 2.5: Hough Transform straight edges overlay on top of the combined edge response map. The purple line represents the straight edge receiving the highest number of HT votes.

Figures 2.8(a) and (b) demonstrate incorrectly estimated pose when using only corner vertices and the correct estimated pose when using inflated edge points respectively.

2.3 Image Features

Many handcrafted image features for target tracking and object recognition were developed over the past four decades. A handcrafted image feature is a keypoint and a descriptor signature. The image keypoint provides the coordinates of an image corner or a visually robust image blob. The image descriptor provides the pattern signature around the keypoint. A good image feature can locate the same point on the target object with the image undergoing multitudes of changes, such as changes in the scale, rotation, viewpoint, illumination, noise, compression and image blur. Many keypoints are built with similar design roadmaps to achieve invariance but use different image analysis techniques. We test the performance of 15 image features from the OpenCV 3 software library*. Some of these image features contain both keypoint locator and feature descriptor, some are only locators, and some are only descriptors. Table 2.1 provides the name and designation of these image features and their references. In the following sections, we provide an overview of each image feature and show performance comparisons using all combinations of detectors and descriptors.

*<https://opencv.org>



Figure 2.6: Edge detection of a difficult to process thermal image. The image contains a heat source on the upper left corner sending heated air towards the center of the image. A model spacecraft is in the lower-mid right side of the image. The first image on the left is the original image; the centre image shows Canny edges extracted from the raw input image; on the right is the HT edge computed from HER and combined edge generation techniques. The third image on the right shows points extracted from the spacecraft model where as the middle image shows only edges from the heat source is detected.

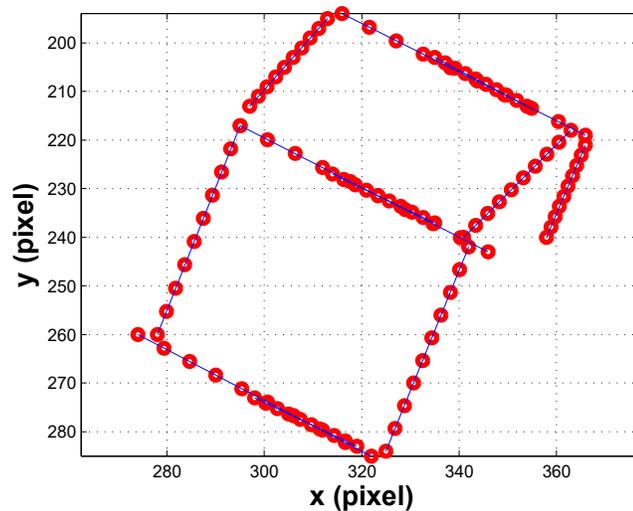


Figure 2.7: Inflating points along the HT straight lines.

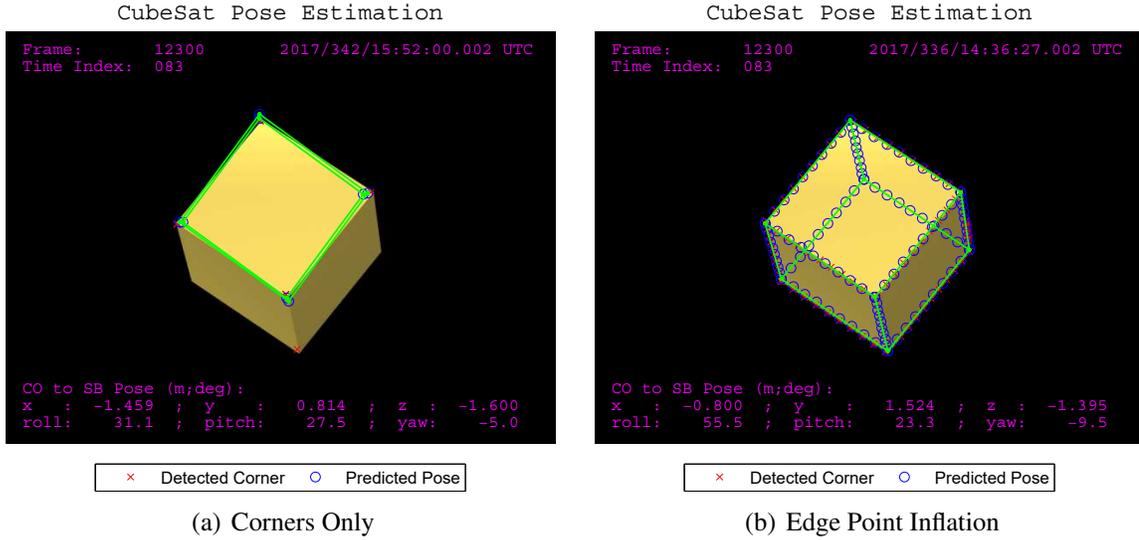


Figure 2.8: Low resolution CubeSat PnP test. Compare corners vertices with edge inflated intermediate points.

SIFT

Lowé [15,333] introduced one of the most widely used image feature called Scale-Invariant Feature Transform (SIFT). The SIFT image feature consists of an image blob extrema keypoint and a bio-inspired image descriptor vector. The SIFT feature remains invariant under scale and rotation changes; it can also match points under limited viewpoint changes. The SIFT image feature and its design roadmap is the de facto benchmark and template respectively for newly developed feature detectors and descriptors.

The SIFT keypoint uses the linear heat diffusion principle to approximate the Laplacian of Gaussian (LoG) with the Difference of Gaussian (DoG) [334] kernel as follows,

$$G(x, y, k\sigma) - G(x, y, \sigma) \approx (k - 1) \sigma^2 \nabla^2 G, \quad (2.3)$$

where G is the Gaussian kernel, σ is the Gaussian kernel scale, and k is some constant incrementing the Gaussian scale. The SIFT feature consists of a layered pyramid [335] for finding features of different scales; this allows the SIFT feature to be scale invariant. Each pyramid layer is called an *octave*, and within each *octave*, we compute the DoG response maps of various scales. The maxima and minima [336] of DoG response map compared to neighbouring pixel and scale layers are selected as keypoints [15]. Keypoints

Table 2.1: List of image feature models used for comparison. The code number represent the year of publication.

Code	Description	Ref.	Detector	Descriptor
SIFT 04	Scale-Invariant Feature Transform	[15]	Yes	Yes
SURF 06	Speeded-Up Robust Features	[14]	Yes	Yes
ORB 11	Oriented FAST and Rotated BRIEF	[233]	Yes	Yes
AKAZE 13	Accelerated-KAZE	[10]	Yes	Yes
GFTT- Harris 88	Good Features To Track Harris Corner	[229]	Yes	No
GFTT-Shi- Tomasi 94	Good Features To Track Shi-Tomasi Corner	[325]	Yes	No
MSER 02	Maximally Stable Extremal Regions	[326]	Yes	No
FAST 06	Features from Accelerated Segment Test	[230]	Yes	No
AGAST 10	Adaptive and Generic corner detection based on Accelerated Segment Test	[327]	Yes	No
Star- CenSurE 11	Center-Surround Extrema Maximum	[328]	Yes	No
MSD 14	Self-Dissimilarities	[329]	Yes	No
DAISY 10	DAISY Pattern Descriptor	[330]	No	Yes
BRIEF 10	Binary Robust Independent Elementary Features	[13]	No	Yes
BRISK 11	Binary Robust Invariant Scalable Keypoints	[12]	No	Yes
LUCID 12	Locally Uniform Comparison Image Descriptor	[331]	No	Yes
FREAK 12	Fast REtinA Keypoint	[11]	No	Yes
LATCH 15	Learned Arrangements of Three patCH codes	[332]	No	Yes

that are low contrast and therefore less stable under noise is eliminated by comparing the extremum DoG response map with some threshold. The extremum DoG can be computed using the Tyler expansion approximation [15]. To further stabilise the SIFT keypoints,

edge responses are eliminated using Harris eigenvalue approach [229] by constructing a Hessian matrix of principal curvatures. Keypoints that does not satisfy the Hessian matrix trace to determinant ratio are removed. The keypoint orientation is also computed so the feature descriptor can be aligned when matching keypoints; this allows the SIFT feature to be rotationally invariant. The keypoint orientation is the dominant angle in the orientation histogram of 36 bins covering 360 degrees based on the inverse tangent of the x and y pixel gradient in the Gaussian smoothed image [15]. Around each keypoint, an orientation histogram of 8 directional bins summarize the contents over a 4×4 subregion; this results in a 128 element descriptor vector. The SIFT feature descriptors are inspired by visual cortex complex neuron response to the gradient at a particular orientation and spatial frequency [15].

Figure 2.9 shows the SIFT features extracted from an ISS *TriDAR* thermal image. The scale and orientation are represented by the size and angle shown on the keypoint circles. The SIFT feature is robust and has been widely used in academia and industry; however, SIFT is computationally expensive due to the many steps involved in the keypoint and descriptor computation.

SURF

Speeded Up Robust Features (SURF) [14] follows the SIFT design template and is also scale and rotationally invariant; it uses more efficient image analysis techniques to achieve the same objective as SIFT. SURF approximates the Gaussian with the faster box filter; it collects interest points based on the determinant of the Gaussian response second order derivative, the so-called Hessian matrix [14]. To achieve scale invariance, the SURF image pyramid up-scales with constant cost by taking advantage of the box filter's four-point integral technique [14]. To achieve rotational invariance, a sliding orientation window of 60 degrees detects the dominant Haar wavelet around every keypoint [14]. The SURF descriptor is a 64 element feature vector based on the sum of the Haar wavelet responses in vector and magnitude and builds an oriented quadratic grid similar to SIFT [14]. The SURF image descriptor is faster than the SIFT feature; however, it is still more intensive in memory and computation compared to binary descriptor techniques.

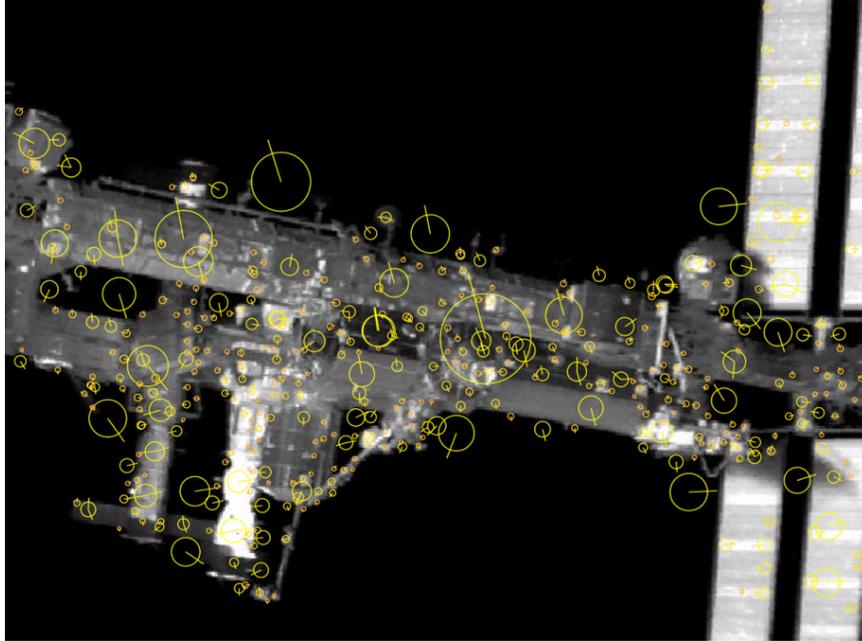


Figure 2.9: SIFT features of the ISS thermal image. Each circle represents a keypoint position; the lines in the circles represent keypoint orientation, and the circle size represents the keypoint scale.

ORB

Both SIFT and SURF keypoints are image blob extremes; alternatively, keypoints may also be corners, such as Harris [229, 337], or Features from Accelerated Segment Test (FAST) [230]. The FAST corner is an enhancement of the *Smallest Univalve Segment Assimilating Nucleus* (SUSAN) corner detector [338]; it selects the corner keypoint by performing a pixel brightness test in the surrounding circle. FAST has been demonstrated to be faster than both Harris corners and SUSAN [339]. Binary descriptors store the contrast signature surrounding the keypoint using binary difference patterns instead of decimal-based numbers. The binary format reduces memory storage and can be computationally efficient using the Central Processing Unit (CPU) XOR or bit count operations. For example, the Binary Robust Independent Elementary Features (BRIEF) descriptor [13] uses Gaussian distributed random points for the binary descriptor vector; more details of the BRIEF descriptor is provided in Sec. 2.3 The Oriented FAST and Rotated BRIEF (ORB) [233] combines the FAST corner and the BRIEF descriptor. Alternatively, Harris [229] or Shi-Tomasi [325] corners have been used instead of the FAST corner; more details on the

Harris [229] and Shi-Tomasi [325] corners are provided in Sec. 2.3. ORB achieves scale invariance by computing the image pyramid [340]; and achieves rotational invariance by calculating the keypoint orientation using intensity centroid of corner moments [341]. The ORB BRIEF descriptor may also undergo greedy search learning for higher precision [233]. The default BRIEF and ORB binary patterns from OpenCV 3 are shown in Figs. 2.15(a) and (b) in Sec. 2.3.2 respectively.

AKAZE

Alcantarilla *et al.* introduced the Accelerated-KAZE (AKAZE) image feature [10] as a faster enhancement of the KAZE feature [342]. Accelerated-KAZE follows the SIFT design template of keypoint detection, scale and rotation invariance, and descriptor generation. Alcantarilla *et al.* replaces the DoG linear diffusion approach with the nonlinear anisotropic diffusion formulation [343] which maintains the natural image edge boundaries [10]. The classic nonlinear diffusion equation is

$$\frac{\partial \mathbf{L}}{\partial \sigma} = \text{div}(c(x, y, \sigma) \cdot \nabla \mathbf{L}), \quad (2.4)$$

where div and ∇ are the divergence and gradient operator respectively. L is the image luminance, and c is the so-called *conductivity* function [343]. x, y represent the image coordinate, and σ is the scale. The KAZE image feature uses a so-called Additive Operator Splitting (AOS) [342]; AKAZE replaced AOS with Fast Explicit Diffusion (FED) [10] to solve the anisotropic diffusion problem iteratively. The AKAZE feature builds the nonlinear scale space pyramid similar to SIFT, and uses the image luminance Hessian determinant for keypoint detection similar to SURF. Rotational invariance is obtained by finding the dominant orientation in a circular area by testing the derivative vector of the luminance response [342]. The AKAZE feature uses Modified-Local Difference Binary (M-LDB) as the feature descriptor. The Local Difference Binary (LDB) [344] descriptor computes binary differences similar to BRIEF but uses binary test between the average local areas instead of individual pixels. The M-LBD uses the luminance derivative, considers the patch rotation, and replaces the local averaging of LBD with sub-sampling depending on the scale [10].

Corners

Corner detection methods have been developed over the past four decades to be more robust under lighting, rotation, scale, and affine viewpoint variations. Harris and Stephens [229] introduces one of the earliest image corners; it is detected when iso-response contours of the determinant and square of the trace to a squared difference matrix, the so-called *structural tensor*, changes direction in the eigenvalue space. Due to the eigenspace rotation, the Harris corner is invariant to the rotation but is not invariant to scale or affine changes. Shi and Tomasi [325] improves the Harris corner by defining the iso-response contour metric; it is the minimum absolute value of the structural tensor eigenvalues. The Shi-Tomasi corner [325] is more robust than the Harris corner [229] when the image undergoes affine transformations [345]. Both Harris and Shi-Tomasi methods are provided by OpenCV 3 under the Good Feature To Track (GFTT) function.

MSER

Region centres can be used as keypoints. Matas *et al.* proposed Maximally Stable Extremal Regions (MSER) [326] by introducing a set of regions closed under continuous and monotonic image intensity transformations. The measurement regions are selected at multiple scales of some distinct region that are one and a half, two, and three times scaled convex hull of that region [326]. The MSER regions are invariant to affine transformation and allows multi-scale detection [346]. The external regions can be enumerated in $O(n \log(\log(n)))$, where n is the number of pixels in the image.

AGAST

Adaptive and Generic corner detection based on Accelerated Segment Test (AGAST) [327] is a variation of the FAST corner [230]. The Accelerated Segment Test (AST) corner must have S connected pixels on the surrounding region which is brighter or darker than some threshold of the central pixel value. The AST applies a minimum brightness difference threshold when comparing a pixel with the surrounding pattern; it uses a binary decision tree to search from pixel to pixel. AGAST builds two trees and specialises homogeneous and structured regions based on small and large values of similarity probability

to the nucleus; it switches between the two trees adaptively when the pixel neighbourhood changes [327].

Star (CenSurE)

The Centre-Surround Extrema (CenSurE) feature [328] is a bi-level normalized approximate LoG using Haar wavelets at different scales. The CenSurE feature could be a circle, octagon, hexagon, or box inner and outer regions that are either a light area surround by a dark one or vice versa. CenSurE uses seven scales with block size between one and seven and finds the extrema by comparing each point in 3D image-scale space with 26 neighbours in scale and position. The result is a faster approximation of the LoG than DoG. The Star keypoint is derived from CenSurE; it is part of the OpenCV 3 vision library.

MSD

Tombari and Stefano introduced Maximum Self-Dissimilarities (MSD) [329] keypoints. The MSD detector is the local maxima Non-Maxima Suppression (NMS) of the Contextual Self-Dissimilarity (CSD) image patches. The CSD image patches are computed based on the dissimilar measures between the image patch and the nearby ones in a relatively large surrounding area. A canonical orientation is associated to each keypoint by accumulating into an orientation histogram between the interest point and centre of the most similar patches weighted by their dissimilarity; this provides the direction corresponding to the highest bin in the histogram [329].

DAISY

The DAISY descriptor [330] computes Gaussian convolved orientation maps for quantized directions and scales. Each descriptor composes of eight over lapping circle regions in three-ring layers outward from the centre circle giving it a daisy flower shape. The circle radiuses are proportional to the standard deviation of the Gaussian kernels. DAISY's circular grid has better localization performance than square grids such as SIFT. The DAISY descriptor is calculated by rotating the sampling grid. An essential advantage of circular design and using isotropic kernel is the convolved map does not have to be recomputed under rotation.

BRIEF

The binary descriptor usually requires less memory storage than a decimal one. The BRIEF descriptor [13] is a binary vector formed from the intensity difference of paired points in a smoothed patch around the keypoint. The test τ is defined on a patch \mathbf{p} of size $S \times S$, τ is 1 if $\mathbf{p}(\mathbf{x}) < \mathbf{p}(\mathbf{y})$ and 0 otherwise; $\mathbf{p}(\mathbf{x})$ is the pixel intensity of the smoothed patch at the image coordinate \mathbf{x} , where \mathbf{x} and \mathbf{y} are n numbers of randomly selected pairs. The BRIEF descriptor is defined as an n -dimensional bitstring $f_n(\mathbf{p}) := \sum_{i \in n} 2^{i-1} \tau(\mathbf{p}; \mathbf{x}_i, \mathbf{y}_i)$. The patch size S is set to 31 [233] with Gaussian smoothing kernel patch of 9×9 and the Gaussian scale set to $\sigma = 2$. Calonder *et al.* [13] tested different sizes of n and found 256 to be the optimal, he used BRIEF- k to denote the group size, where $k = n/8$ representing the number of descriptor bytes. Tests of various point pair patterns show that an independent and identically distributed random Gaussian distribution is superior to a polar pattern and the uniformly randomized distribution. A Gaussian BRIEF pattern of scale $\sigma = S/r = 31/5$ is shown in Fig. 2.14(a). Rublee *et al.* [233] further improves the BRIEF pattern by rotating it by the corner orientation and applying the greedy search optimization. The rotated BRIEF pattern is designated as steered-BRIEF (*s*BRIEF), and the optimized BRIEF pattern is called *r*BRIEF. The advantages of a randomized binary pattern over an ordered one is the basis of the new descriptor discussed in Sec. 2.3.2.

BRISK

Leutenegger *et al.* proposed Binary Robust Invariant Scalable Keypoints (BRISK) [12] to address BRIEF's weakness in rotational variations. The BRISK descriptor is a binary string by concatenating the results of simple brightness comparison tests; it consists of circular rings that resemble the DAISY descriptor [330]. BRISK feature pairs can be categorised into short and long-distance subsets. The long-distance subset computes normalized image intensity difference, and then it compares between the same angle pair intensities within the short-distance subsets, and finally computes the binary output for each pair. To achieve rotational invariance; the BRISK angle patches are calculated by the *arctan* of the local gradient [12].

LUCID

Locally Uniform Comparison Image Descriptor (LUCID) [331] is a grid binary descriptor patch that compares the Hamming distance of two sorted image patches. The LUCID feature can be used in colour images by taking the patch in each channel then combine them into one sorted array. The patch comparison can be summarised in three lines of MATLAB code as follows:

```
[~,desc1]=sort(p1(:));
[~,desc2]=sort(p2(:));
distance=sum(desc1~=desc2);
```

where $p1$ and $p2$ are the two descriptor patches, $desc1$ and $desc2$ is the order permutation, and $distance$ is the Hamming distance between the two sorted descriptor orders [331]. The Hamming distance of the order permutation is the so-called *disorder distance*; it measures the disorder between two permutations and is not sensitive to Gaussian noise [331].

FREAK

The Fast RETina Keypoint (FREAK) [11] is a binary descriptor that is bio-inspired and mimics human retina cell functions. FREAK uses circular patterns similar to DAISY and BRISK with higher overlap. The FREAK patterns resemble Gaussian distribution as cell density reduces going from fovea to parafovea to perifoveal regions in the retina. The fovea region has the smallest area with highest point density and is used to recognize and match objects. The outer perifoveal has the largest area but least point density and is used to capture low-frequency observations; it is used to compile first estimates of the object location. FREAK uses search pattern similar to the saccadic eye movement; it checks coarse patch points for matching. For orientation invariance, FREAK use similar method as BRISK, where only the central area points are used for gradient computation. Similar to ORB, FREAK include unsupervised learning to choose the optimal set of point pairs [11].

LATCH

The Learned Arrangements of Three Patch Codes (LATCH) [332] is an improved Local Binary Patterns (LBP) [347] and Three-Patch LBP [348]. The LATCH descriptor considers multiple arrangements of three matrix descriptor patches and learns the optimal arrangement from the training data. Specifically, the LATCH algorithm evaluates three small randomly generated patches by taking the difference between two patch pairs as follows,

$$g = \begin{cases} 1 & \text{if } \|\mathbf{P}_a - \mathbf{P}_1\|_F^2 > \|\mathbf{P}_a - \mathbf{P}_2\|_F^2 \\ 0 & \text{otherwise} \end{cases} \quad (2.5)$$

where \mathbf{P} is the image patch for region a , 1, and 2, and $\|\cdot\|_F^2$ is the Frobenius norm. By using three patches, LATCH is less sensitive to noise [332].

2.3.1 Feature Comparison

The following section provides a comprehensive comparison of the image features previously described. Tests focusing on precision and computation speed is performed using the standard image feature benchmark Oxford dataset [349].

Test Images and Metrics

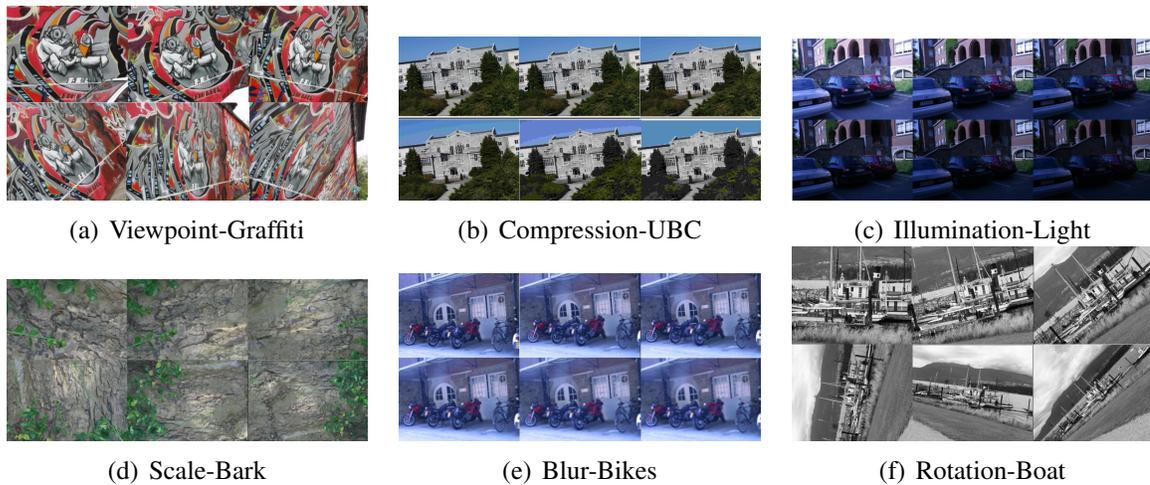
While it is not related to spacecraft imagery, the Oxford dataset [349] is the standard benchmark for image feature development [†]. The Oxford dataset includes six sets of images that is specialised in testing viewpoint, compression, illumination, scale, image blur and rotation invariance. Each set contains six images, from the zero reference baseline and progressively increase in difficulty. The homography matrix for each image is provided for ground truth comparison. Table 2.2 provides details of the Oxford dataset images. Figure 2.10 shows the Oxford dataset image sequences.

Image features were computed on the baseline image and the test images. The features in the two images are matched using a feature matcher. The precision is the number of true matches over all matches. The match error is the homography transformation of the test image matches to the baseline image subtract the baseline keypoint locations. A true

[†]<http://www.robots.ox.ac.uk/~vgg/research/affine/>

Table 2.2: Oxford dataset descriptions.

Sequence	Description	Image Size
Graffiti	Affine viewpoint	800×640
UBC	Compression	800×640
Light	Illumination	921×614
Bark	Rotation and zoom	765×512
Bike	Image Blur	1000×700
Boat	Rotation and zoom	800×640

**Figure 2.10:** Oxford dataset images.

match is when the normalised match position error is less than or equal to 10 pixels. Three matchers were tested, the Fast Library for Approximate Nearest Neighbour (FLANN) [350] matcher using Least Median of Squares (LMedS) for statistical robustness, a Brute Force (BF) matcher using Random Sample Consensus (RANSAC) [94] for robustness, and a k -Nearest Neighbour (k NN) matcher using RANSAC. The FLANN matcher was eliminated because it was not compatible with several descriptors. The k NN method was selected over the BF matcher due to faster computation.

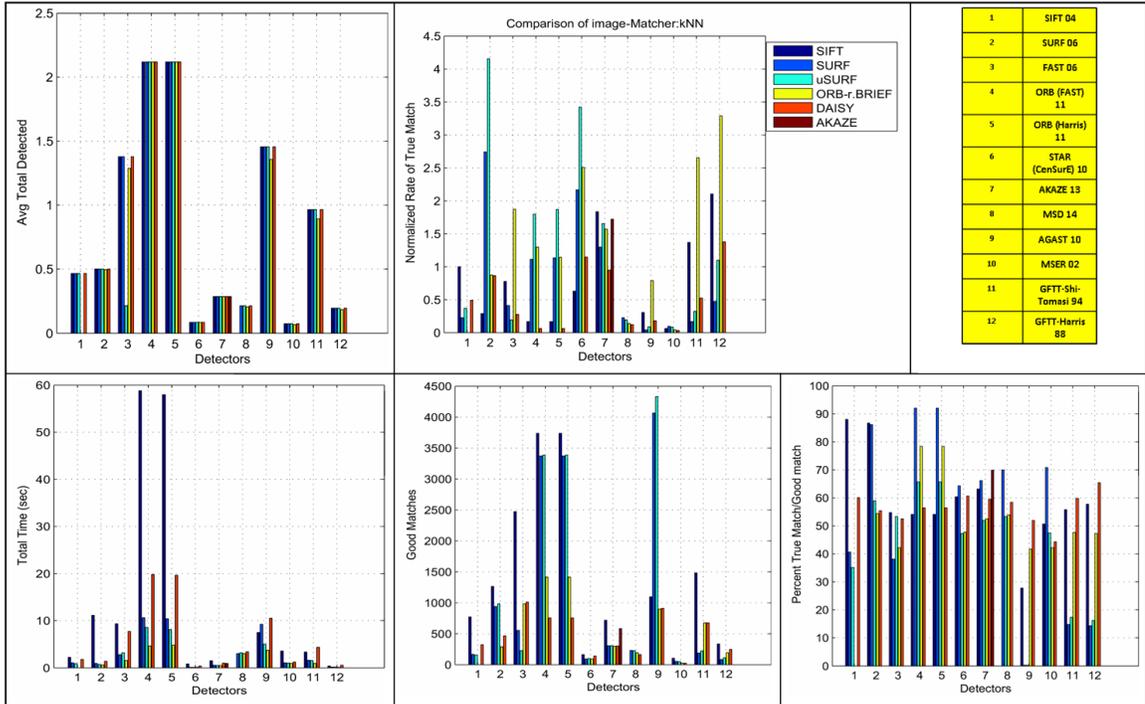
The detector and descriptor introduced in the previous sections were evaluated in all combinations. Table 2.11 provides an overview of the evaluation matrix.

Descriptors		1	2	3	4	5	6	7	8	9	10	11	12
Detectors		SIFT 04	SURF 06	u-SURF 06	ORB (r-BRIEF) 11	DAISY 10	AKAZE 13	LATCH 15	LUCID 12	BRISK 11	r-BRIEF 10	u- BRIEF 10	FREAK 12
	1	SIFT 04	x	x	x		x		x	x	x	x	x
2	SURF 06	x	x	x	x	x		x	x	x	x	x	x
3	FAST 06	x	x	x	x	x		x	x	x	x	x	x
4	ORB (FAST) 11	x	x	x	x	x		x	x	x	x	x	x
5	ORB (Harris) 11	x	x	x	x	x		x	x	x	x	x	x
6	STAR (CenSurE) 08	x	x	x	x	x		x	x	x	x	x	x
7	AKAZE 13	x	x	x	x	x	x	x	x	x	x	x	x
8	MSD 14		x	x	x	x		x	x	x	x	x	x
9	AGAST 10	x	x	x	x	x		x	x	x	x	x	x
10	MSER 02	x	x	x	x	x		x	x	x	x	x	x
11	GFTT-Shi- Tomasi 94	x	x	x	x	x		x	x	x	x	x	x
12	GFTT-Harris 88	x	x	x	x	x		x	x	x	x	x	x

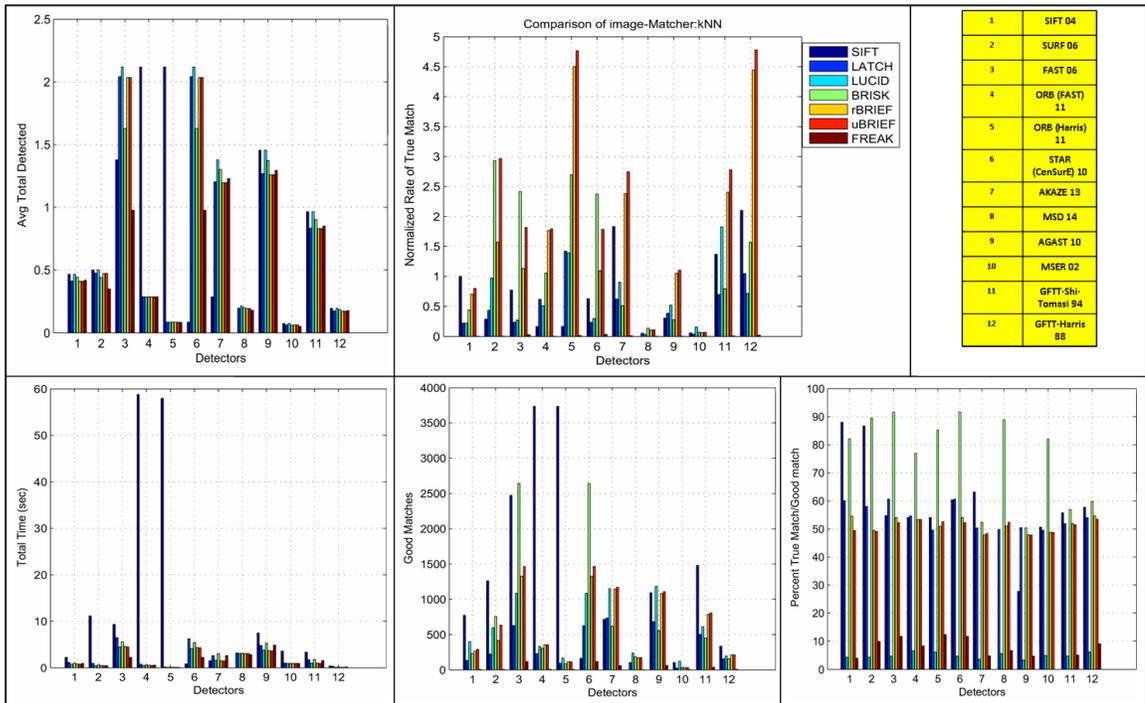
Figure 2.11: Image feature evaluation matrix.

Feature Performance

Results of the feature evaluation are provided in Fig. 2.12. Some results are normalised to the SIFT [15] feature as a ratio for straightforward benchmark comparisons. Figure 2.12 shows ratio comparisons in average total match detected, the normalised rate of true match, total computation time, the total number of matches, and percent precision. The normalised rate of true match is the true match over computation time. On average, image features with the highest precision and best balance between precision and timing are selected. The highest precision feature is using the ORB-FAST corners with the SIFT descriptor. It was surprising after many years of image feature development the SIFT descriptor still holds high precision compared to the newer techniques. The image feature with the best balance in precision and speed is using the Harris corner with the BRIEF descriptor. Although the BRIEF descriptor is best for speed and precision; when applied to space imagery, it still lacks the sufficient precision. We improve BRIEF by introducing a novel feature descriptor called γ BRIEF, where by changing the random binary pattern we increased BRIEF descriptor performance.



(a)



(b)

Figure 2.12: Feature comparison results.

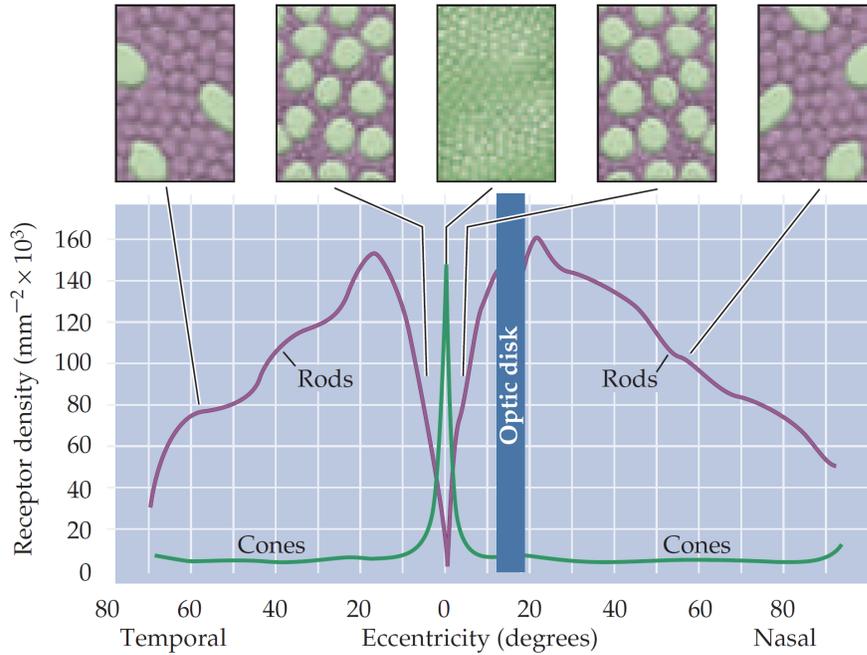


Figure 2.13: Human retina receptor density distribution. Figure reproduced from [1].

2.3.2 *y*BRIEF

We develop a novel descriptor inspired by the randomly generated binary patches of BRIEF [13] and cells in the retina similar to FREAK [11]. We were also inspired by LATCH [332] which uses three coordinates for second order differencing. The bitwise differencing of two Gaussian patterns resembles the *centre-surround* organization of the receptive field. It is well known that the photoreceptors in the retina are made from rods and cones. The rods have low spatial resolution and are sensitive to light and the cones are the opposite. The rod and cone cells have non-Gaussian distributions as shown in Fig. 2.13 [1]. Our random binary descriptor mimics these retina cell patterns.

A mirrored Levy Probability Density Function (PDF) is used to build the rod cells, this is denoted as p_r in the following equation,

$$p_r(x) = \sqrt{\frac{c}{2\pi}} \frac{\exp\left(-\frac{c}{2(x-\mu)}\right)}{(x-\mu)^{3/2}}, \quad (2.6)$$

where μ is the mean and is centred to zero. The patch is set to be twice as large as the

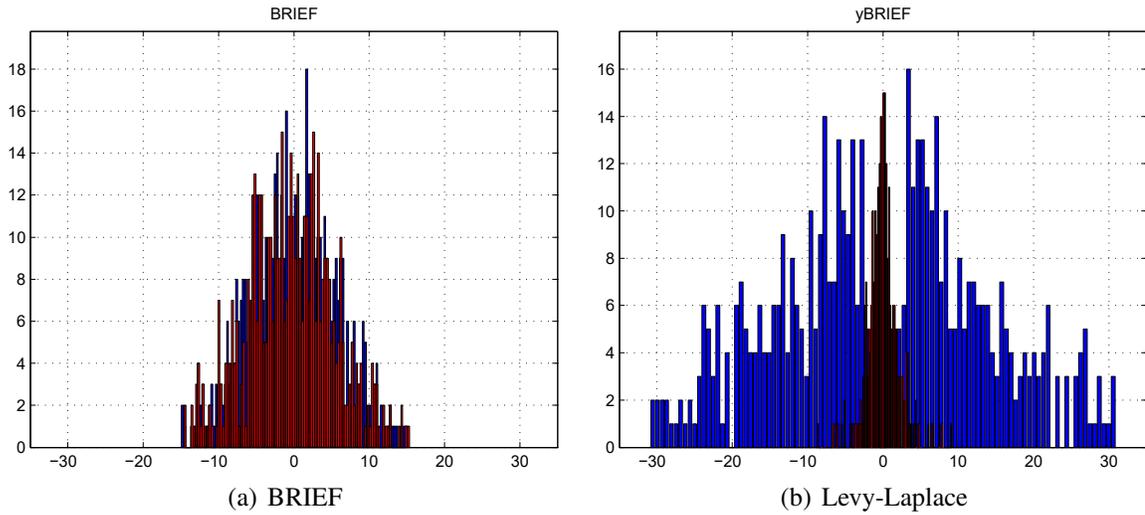


Figure 2.14: Random distribution histograms.

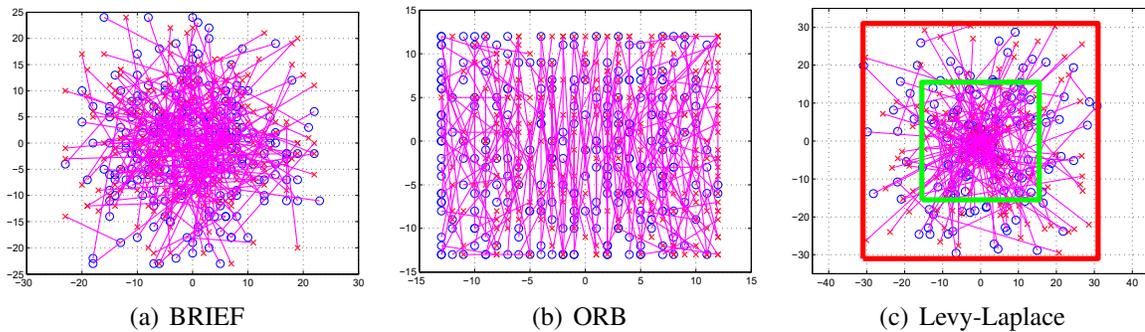


Figure 2.15: 2D binary patterns.

BRIEF patch for best shape *i.e.* $S_y = 2S$, where c is the scale and is tested to be S_y/r , and S is the width of the descriptor patch.

The cones pattern can be represented by using the Laplace PDF, denoted as p_c in the following equation,

$$p_c(x) = \frac{1}{2b} \exp\left(-\frac{|x - \mu|}{b}\right), \quad (2.7)$$

where μ is zero, b is the scale and is manually tuned to be $c/8$. Combining the two distributions in 2D, a 50 percent cones-to-rods ratio Levy-Laplace binary pattern is shown in Figs. 2.14(b) and 2.15(c). Since this descriptor is inspired by cell patterns in the eyes, and is a variation of the classical BRIEF descriptor, y is added to BRIEF and henceforward shall be called the y BRIEF descriptor. The y BRIEF software is provided in Appendix C.3.

Differencing Patterns

The baseline *y*BRIEF pattern uses 256-bit rods and 50 percent cones resulting in 48-bytes. Pattern R2C represent differencing between only rods and cones; pattern IND represent differencing between rods independent from cones; pattern RAND is a random mixture of rods and cones pairing; pattern VERT sorts the x-axis coordinates resulting in vertical differencing lines; this resembles the optimized ORB pattern shown in Fig. 2.15(b). Each of the previously described patterns has its own advantages in different image variations. One way to retain the benefits of each pattern is to combine them into one matrix. Pattern COMB combines R2C, IND, RAND and VERT connections into one large matrix of 192-bytes. While results show COMB has a slightly higher advantage than the other patterns, it is slow to compute and defeats the purpose of memory efficient binary descriptors. Pattern CMBLT and $CMB \times 2$ reduces the full COMB to 12-byte and 24-byte individual pattern equivalents by randomly extracting points from the respective coordinate bins. Pattern HCHV and HCV gives more weighting to the vertical pattern by randomly selecting more points from the vertical pattern bin. HCHV contains half of the combined patterns and half of the vertical patterns randomly selected to a total of 48-byte equivalence. HCV contains half of the combined patterns and full vertical patterns which is 72-bytes. The various rods and cones connection patterns and descriptor sizes are shown in Fig. 2.16. The red boxes are the *y*BRIEF descriptor patch size and the green boxes are the conventional BRIEF descriptor patch size. Performance results for the various patterns are provided in Sec. 2.3.2.

Corner Orientations and Scale Invariance

*y*BRIEF is a variation of the BRIEF and ORB's *r*BRIEF descriptor; therefore, it is naturally more suited for corner keypoints using Harris or FAST scores rather than blob keypoints such as SIFT, SURF, and AKAZE. This property was demonstrated with tests using both corner and blob keypoints. Rotation adjustments for the *y*BRIEF is generated using the corner orientation intensity centroid same as ORB. Scale invariance is improved by generating image pyramids and extracting keypoints from each pyramid layer. To gain a precise understanding of the *y*BRIEF pattern effectiveness, the *s*BRIEF and BRIEF tests shall use the same keypoints as *y*BRIEF. The corner keypoints are generated by using image pyramids,

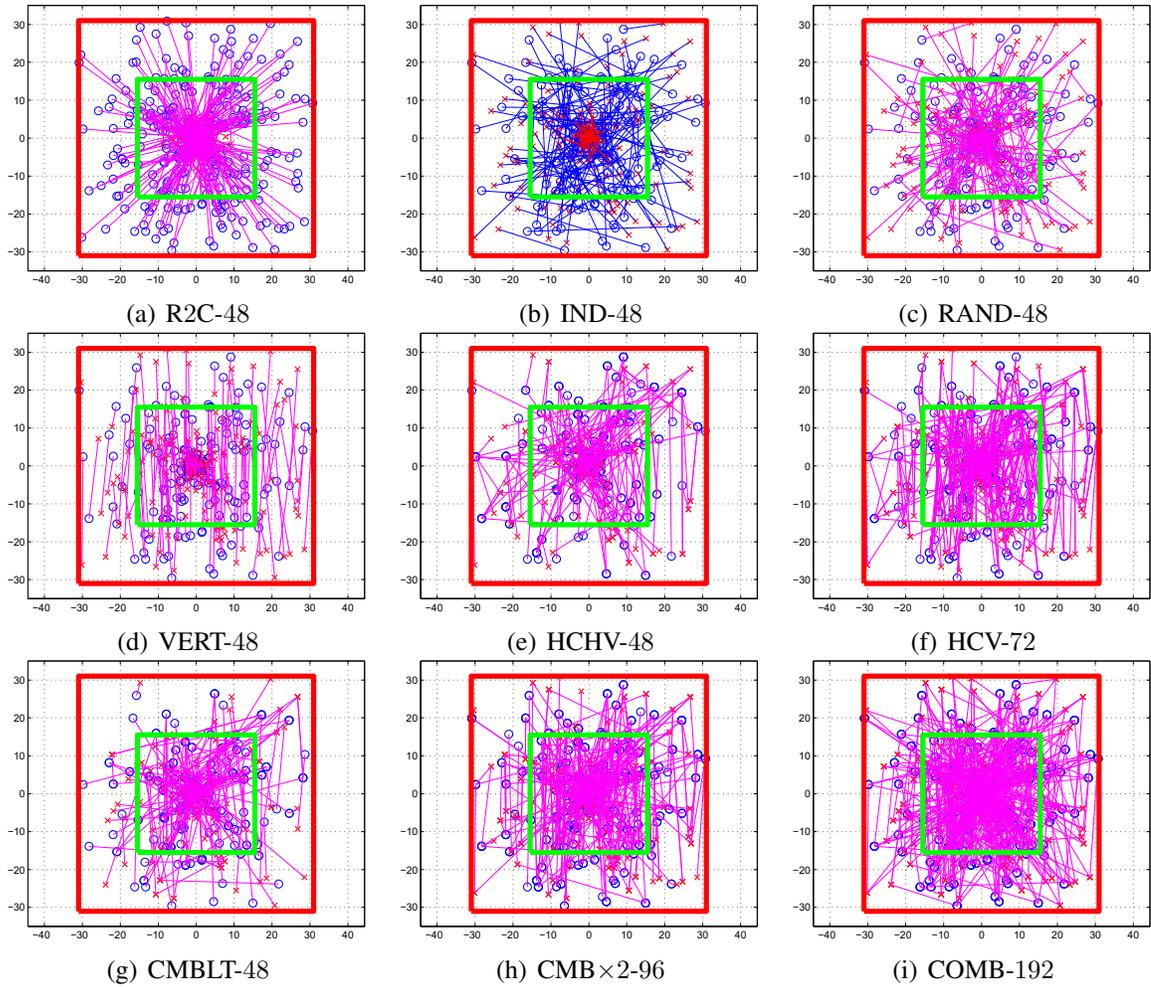


Figure 2.16: y BRIEF test patterns: (o-Start, \times -End).

the number of octave layers is 8 and the scale factor is 1.2 same as ORB.

***Lc* Differencing**

Both BRIEF and ORB perform patch smoothing to reduce the influence of noise on the image. In OpenCV 3, BRIEF uses a box filter, while ORB uses a Gaussian filter for smoothing. Inspired by LATCH [332], we investigate an alternative method to handle image noise, we use a third point to compute second order differencing in addition to the single pair differences. The second order differencing bit is stored with the original descriptor and is called *Lc* differencing. The first two points in *Lc* differencing are a rod and a cone; the number of *Lc* points are selected to be 50 percent of the rods. An example of the 90 degrees *Lc* configuration is shown in Figs. 2.17(a) and (b). The image tests show, however, while more computationally expensive, Gaussian smoothing still outperforms the box filter and *Lc* differencing on average. The results of these tests are given in Sec. 2.3.2.

Experimental Images

The Oxford dataset provided in Sec. 2.3.1 is used for feature comparisons. For robustness to noise, the *Iguazu* image data set [10] is also added to the test matrix. The full image dataset for the *y*BRIEF evaluation is provided in Fig. 2.18.

We use Receiver Operating Characteristics (ROC) [351] to evaluate descriptor performance [349]. The ROC precision, also known as the Positive Predicted Value (PPV) is defined as $Precision = TP/EP$, where EP is the estimated positive or all matches, and TP is the true positive matches from EP . TP is confirmed by the homography transformation of the training keypoint to within 5 pixels of the matched query keypoint. A true non-match requires no transformed training keypoints to fall within the pixel proximity of a query keypoint. The ROC recall is defined as $Recall = TP/AP$, where AP is the actual positive matches overall or the sum of TP and the false negatives (FN); FN is all queries minus non-matches and TP ‡. Finally, the *y*BRIEF descriptor timing is scaled by the BRIEF timing as a non-dimensional parameter.

The second test uses an image captured by an ICI-9320P thermal camera, as shown

‡This study computes AP based on matches to query keypoints rather than region correspondence used by Mikolajczyk and Schmid [352].

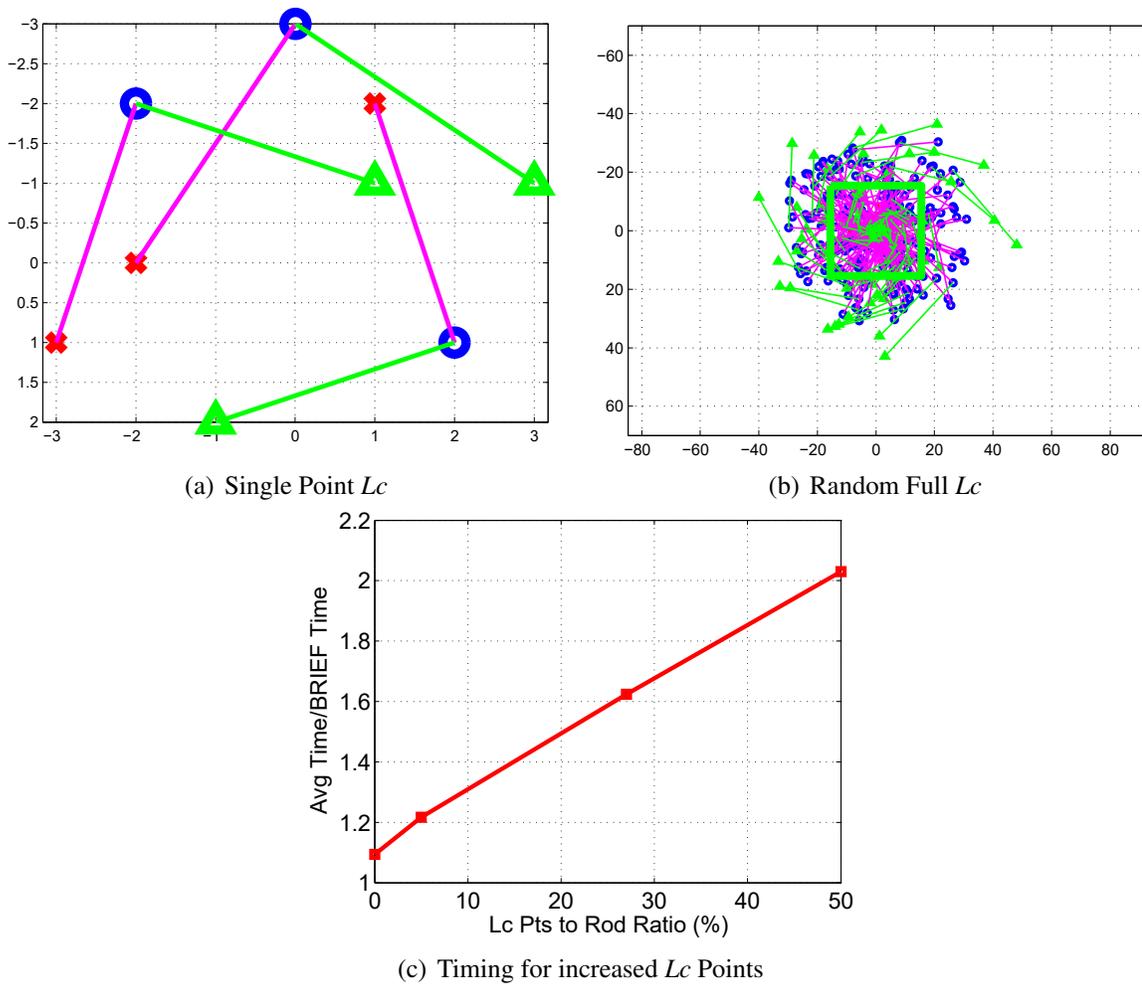


Figure 2.17: L_c -Differencing Patterns: (\times -Cones, o -Rods, Δ - L_c Points)

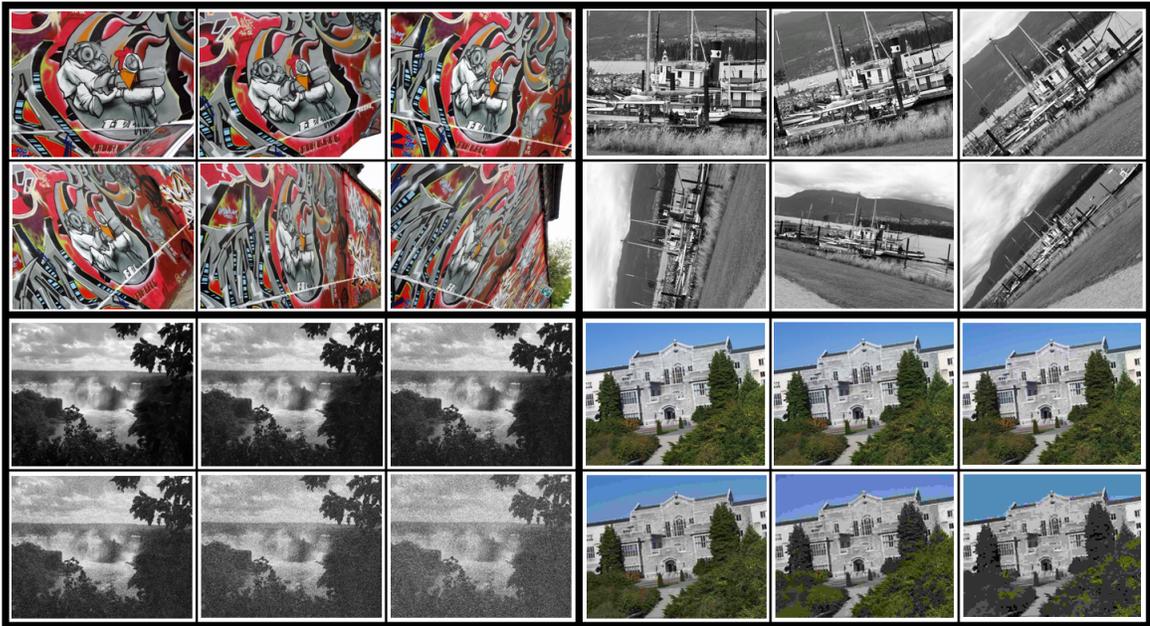


Figure 2.18: y BRIEF test images, Oxford image data set plus Iguazu. Top-left: Graffiti, top-right: Boat, bottom-left: Iguazu, bottom-right: UBC.

in Fig. 2.19(a). The thermal camera resolution is 320×240 and was calibrated using the *Camera Calibration Toolbox for MATLAB*[§]. The thermal image is a room with a *Maneki-neko* figurine^{||} as shown in Fig. 2.19(b). The *Maneki-neko* represent a laboratory tracking target.

Feature Performance

The y BRIEF descriptor is tested against BRIEF, s BRIEF, ORB, AKAZE, SURF, and SIFT. y BRIEF, BRIEF, and s BRIEF are implemented using the code generated in MATLAB while ORB, AKAZE, SIFT and SURF are computed using OpenCV 3 libraries. Descriptor timing is compared between the BRIEF family descriptors. The total number of keypoints for each feature is limited to 500 highest intensity ones. ORB keypoints are computed using the FAST score while BRIEF, s BRIEF, and y BRIEF used the Harris-Noble corner [337] with NMS, corner orientation, and image pyramids. OpenCV 3 k NN routine is used for descriptor matching. SIFT and SURF used the $NORM_L2$ k NN, all of the binary descriptor

[§]http://www.vision.caltech.edu/bouguetj/calib_doc/

^{||}http://www.ai-automata.ca/01_research/0_img/data/Maneki-neko.png

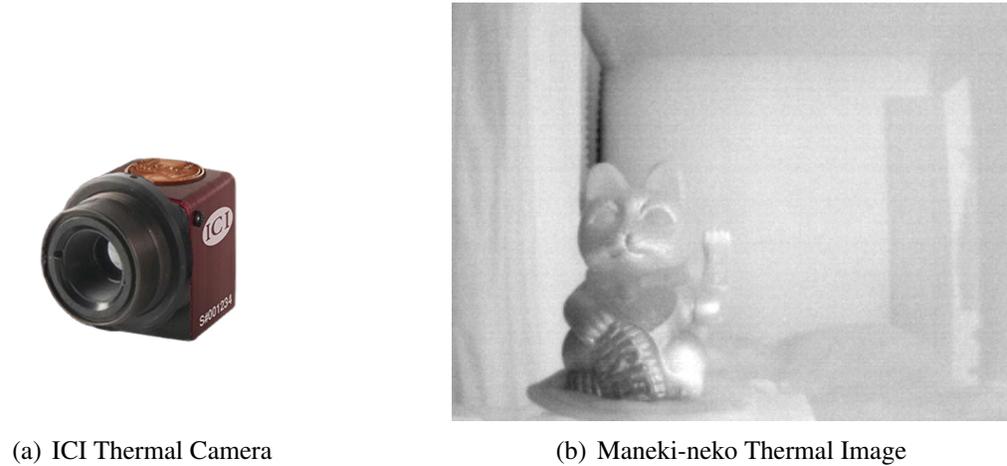


Figure 2.19: Thermal camera and test image.

features used the *NORM_HAMMING2* k NN. For comparable results, the k NN ratio tolerance for SIFT, SURF, AKAZE and ORB is set to 0.875, while s BRIEF and BRIEF is set to 0.95, and y BRIEF is set to 0.925.

Rod and Cone Patterns

Figure 2.20 shows recall versus 1-precision for all the differencing patterns. The COMB pattern slightly outperformed the other patterns in rotation, noise, and pixelation. RAND, however, outperforms COMB in Graffiti where the viewpoint is changed. Furthermore, RAND outperforms all of the patterns with the same descriptor length. Given the higher memory and processing required for longer descriptors, RAND offers the best value of all patterns.

Lc Differencing and Patch Smoothing

Eleven Lc differencing cases were tested on all image sets. These cases are: Lc -off with box filter (L0FB); Lc -off with Gaussian filter (L0FG); Lc -90 deg with Gaussian filter (L90FG); random Cone-Rod-Cone Connections (CRCC); random Rod-Rod-Rod Connections (RRRC); Lc -0 to Lc -180 degrees (Lc - XX); and no latch and no patch smoothing (L0F0). The recall versus 1-precision graphs for the various Lc differencing and patch smoothing techniques are shown in Fig. 2.22 for the four test image sets. Surprisingly, Lc differencing did not have any effect for Graffiti, Iguazu, and UBC. The Lc -45 outperformed the other Lc angles in Boat. Figure 2.17(c) shows with the increasing number of Lc point-to-rod ratio, the computation time increases linearly. While small increases in the

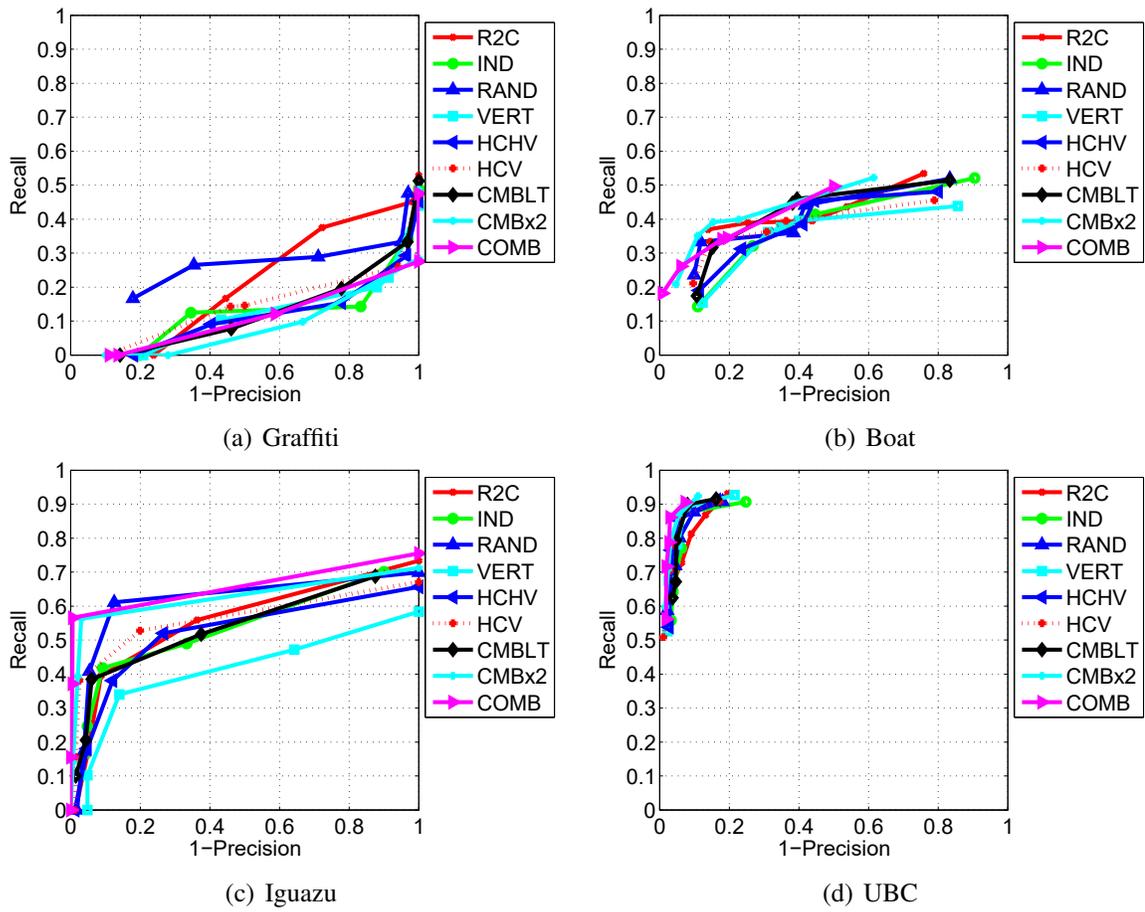


Figure 2.20: Recall vs. 1-precision pattern performance.

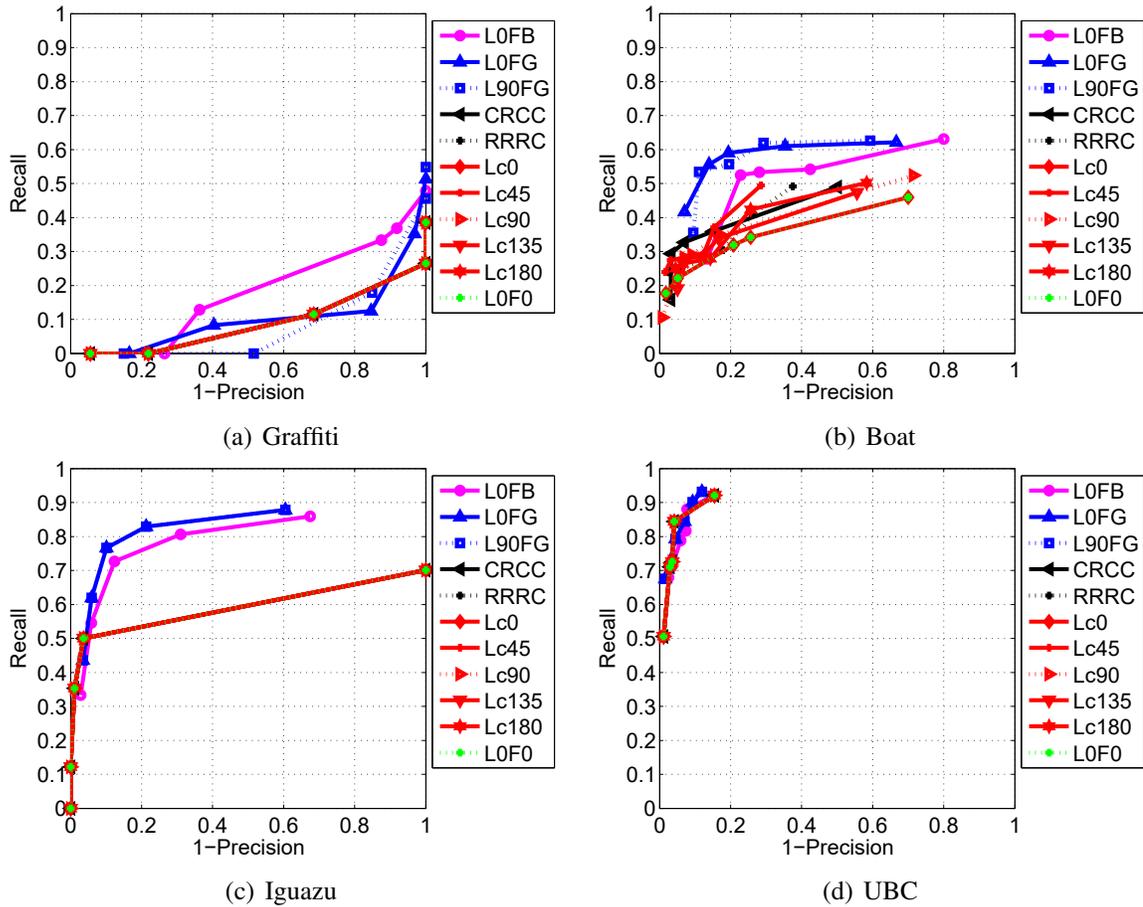


Figure 2.21: Recall vs. 1-precision patch smoothing.

recall have been added by using this technique, it did not amount to significant value for the overall descriptor quality. The Gaussian filter has higher recall and precision than the box filter and can be considered to outperform all methods overall.

Descriptor Performance

In the final test, all features are tested with y BRIEF using the RAND pattern and Gaussian filtering. The recall versus 1-precision graphs are shown in Fig. 2.22. The y BRIEF descriptor outperforms the classical BRIEF and s BRIEF with wide margin, and as indicated by Fig. 2.17(c), there is only a 10 percent increase in the computation time. y BRIEF outperforms ORB and SIFT in Iguazu and UBC respectively. In Graffiti, y BRIEF achieves better recall than AKAZE but lags in precision. In general, y BRIEF performance is tantamount to the standard features but does not lead them.

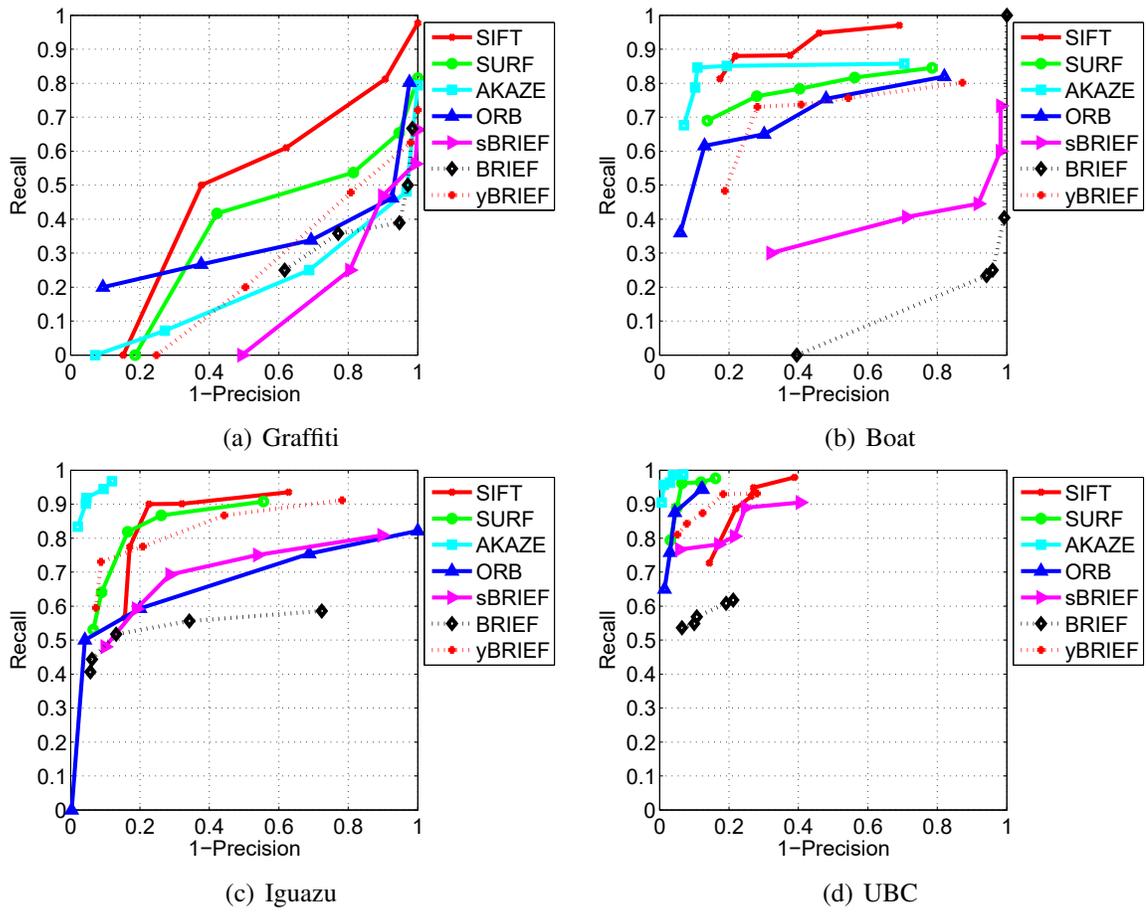


Figure 2.22: Recall vs. 1-precision for all features.

Image Rotation

All binary and non-binary features including SIFT, SURF, ORB, AKAZE, BRIEF, *s*BRIEF and *y*BRIEF were evaluated in the thermal image *Maneki-neko* test, with features limited to 500 of the highest intensity keypoints. Test images were produced by in-plane rotation of the reference image over 360 degrees and computing the homography using the rotation matrix and the calibrated camera properties. The match precision is provided in Fig. 2.23 and the number of true matches is provided in Fig. 2.24. SIFT and AKAZE are steady over all angles where SURF, ORB, *s*BRIEF and *y*BRIEF have larger variations under rotation. ORB has the highest true matches out of all features. This is because ORB in OpenCV 3 does not perform NMS on the FAST keypoints. In the ORB feature, 8 octave pyramid layers of keypoints are stacked on top of each other with many redundancies. Figure 2.25 shows the difference between ORB, AKAZE and *y*BRIEF for a 135 degrees image rotation. Over the same image region, ORB generates greater numbers of keypoints and matches but many are redundant.

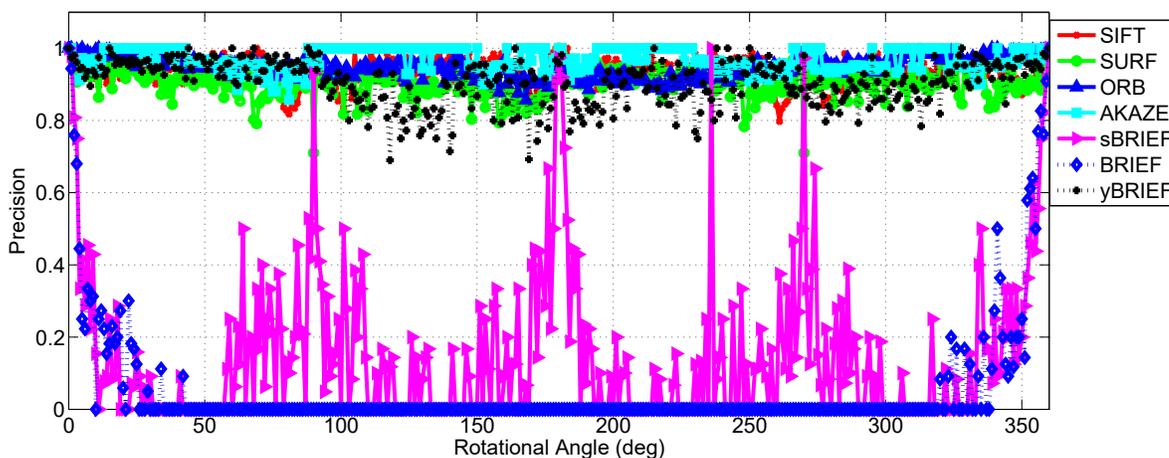


Figure 2.23: Match precision comparisons.

Concluding Remarks

This section developed and tested a novel binary descriptor called *y*BRIEF based on the retina's rods and cones cell patterns. Different variations of the *y*BRIEF patterns are tested in addition to patch smoothing and second order differencing. The *y*BRIEF descriptor performance exceeds the classical BRIEF and *s*BRIEF descriptors with a slight increase in

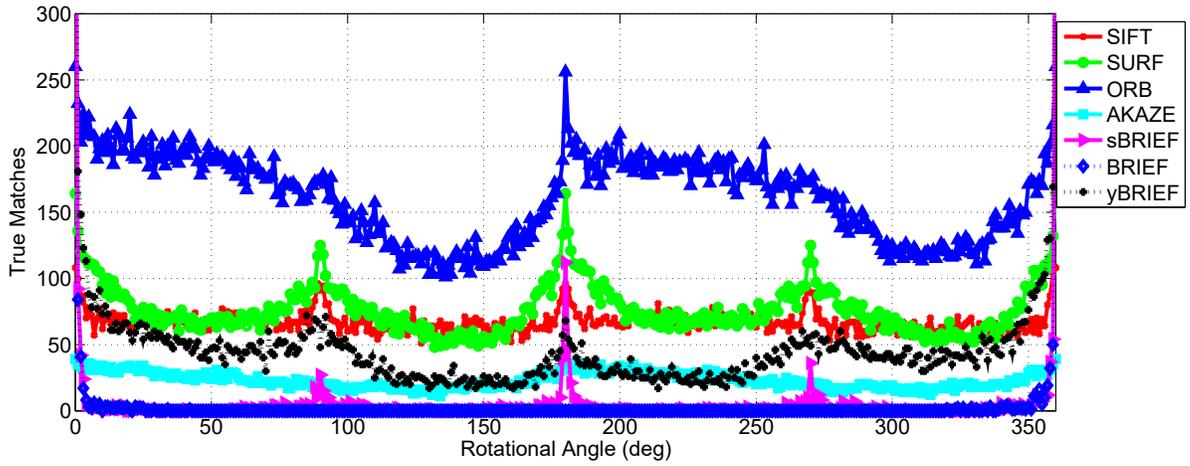
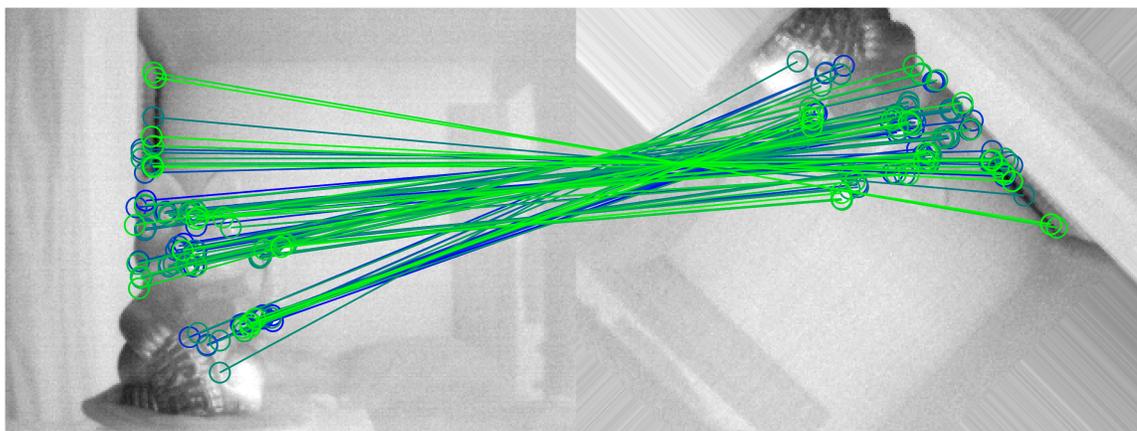
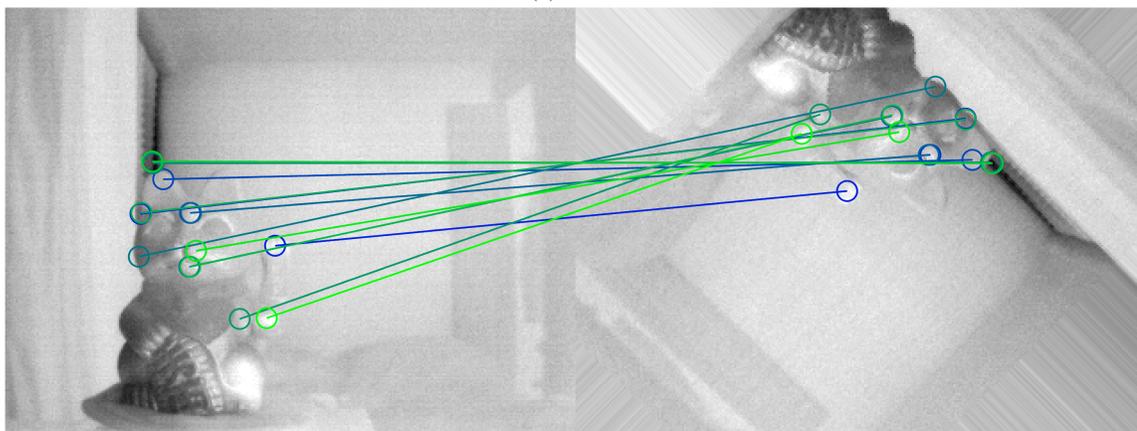


Figure 2.24: True match comparisons.

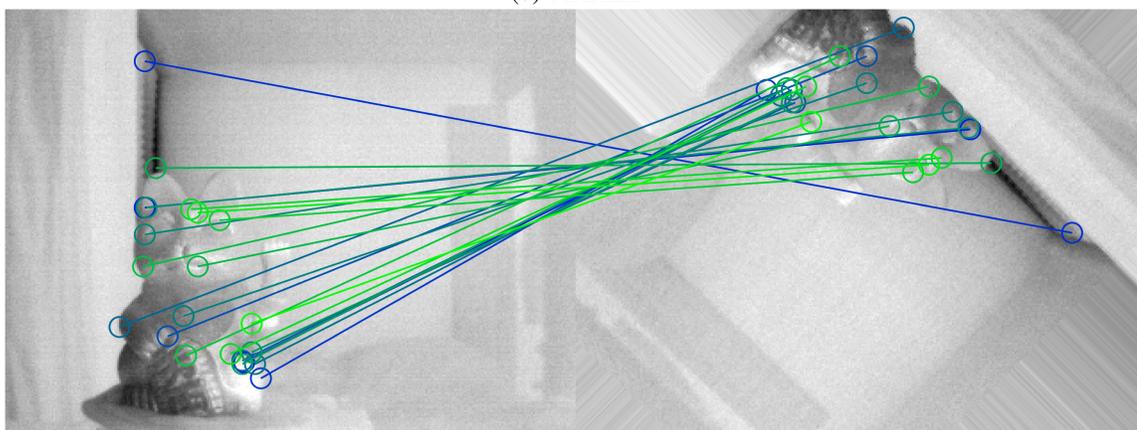
computation time. Finally, the γ BRIEF descriptor performs in tantamount to SIFT, SURF, AKAZE, and ORB.



(a) ORB



(b) AKAZE



(c) yBRIEF

Figure 2.25: Feature matching comparison at 135 deg.

Chapter 3

Target Localisation

Real-world space imagery faces two main challenges, a lack of information due to occlusion from shadows, and confused information due to background clutter or distortions. Distortions may be caused by blurring from fast camera motion or hardware imperfections; in our experience, these are rare occurrences. Out of all the challenges, background clutter is more difficult to overcome even with internal model guidance. Furthermore, processing the entire image including the clutter is computationally wasteful. One way to remove the unuseful background is to recognise and extract the target from the image. For example, the HT straight line extraction introduced in Chapter 2 Sec. 2.2.1 is assuming an artificially made target spacecraft should exhibit more straight lines than the natural background. Indeed, the straight line assumption is a rudimentary one; more sophisticated recognition may take advantage of machine learning and deep learning techniques. To this end, this chapter investigates into using bounding box recognition and localisation of the target object to minimise background clutter from the image.

3.1 Bag of Visual Words

The Bag of Visual Words (BoVW) bounding box technique stems from the Bag of Words (BoW) document retrieval approach that classifies documents based on the likelihood of the *dictionary* words' appearance. Target spacecraft components made from different materials will have different image *textures*. These textures can be grouped in the same way as vocabulary words to facilitate target classification. Since many spacecraft share the same construction materials, for example, the solar panel or multi-layered thermal insulation covering. The *learned* material texture vocabulary can be reused on different vehicles. The BoVW method implements a training phase where the image texture dictionary and component vocabulary are constructed from a database of target spacecraft images, and a classifier learns each component's signature; this is followed by a inference phase, where the image

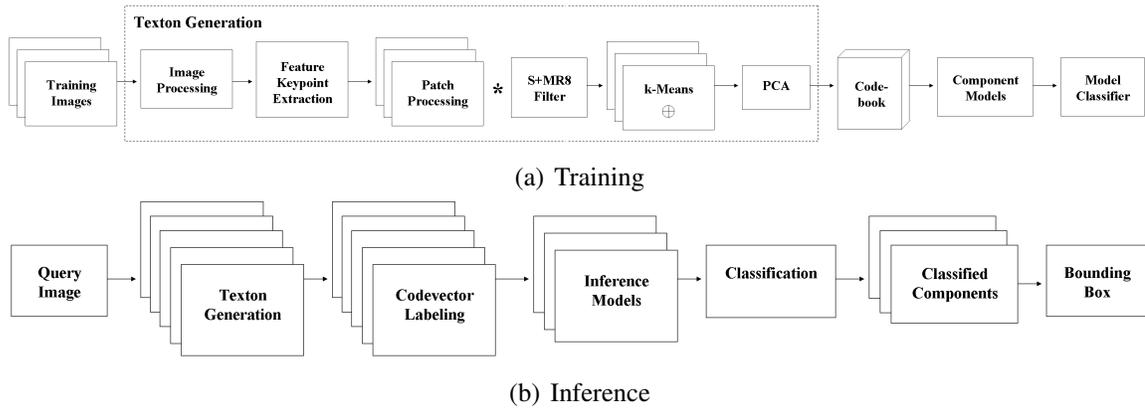


Figure 3.1: Object recognition training and inference process.

features and textures are extracted from the operation image, and the classifier identifies the local region. An overview of the BoVW training and query process is provided in Fig. 3.1.

During the training phase, images of various components are labeled and read into the software. Distinctive features are extracted where the feature keypoints locate a patch window of some predefined size. Each patch image is normalised and convoluted with the Schmid and Maximum Response filters (S+MR8). The filter response maps are clustered using a k -Means algorithm to produce the list of texton images; these *texton* images are converted into the eigenspace via the PCA process. The PCA textons are saved as *codevectors* in the codebook dictionary. The conversion eigen transformation vector (matrix of eigenvectors) is also saved for the inference process. Component models can be formed using the codebook vocabulary texton images. These component models are represented as normalised histograms or feature vectors of unit length. A component model classifier can be developed using linear, non-linear, or non-parametric methods. This classifier will be used to recognise and localise spacecraft components during the inference phase.

In the inference phase, the inference image undergoes the same filter response map generation as the training phase. Each keypoint feature produce a set of filter response maps and is transformed into the eigenspace where the maximum component is matched and labeled with the nearest codevector. The labeled response maps forms the inference model that can then be classified by the previously trained model classifier. Finally, the boundary component keypoints forms the bounding box for the target component. The BoVW software is provided in Appendix C.4.

3.1.1 Textons

Textons are the basic unit representation of material texture [112]. Mean value texton patches from the training images forms the ‘words’ of the *codebook dictionary*. Combinations of texton images build the ‘sentence’ models analogous to the BoW scheme. The following sections provide details of texton generation.

Image Processing and Feature Extraction

The input image is converted to grayscale and normalised. Local patches are extracted to form the codebook vocabulary. FeiFei [118] evaluated evenly sampled grids, random sampling, Kadir and Bradley saliency detector [353], and the SIFT detector [15] to determine the patch position. The even division grid produces the best performance; however, this results in a large number of sample patches and is inefficient to process. The evenly grid patch inefficiency is especially true in many space images where most of the background is empty. Instead, our method uses feature keypoints to position local patches. The keypoints can be computed by SIFT [15] or AKAZE [10]. Alternatively, corner keypoints such as Harris [229] or FAST [230] may be used. We recommend the use of the blob keypoints in cases where a majority of the image is empty. We baseline the SIFT keypoints due to its stability under varying illumination and viewpoint conditions and invariance to rotation and scale. To improve computation efficiency, NMS was applied with the patch size as the areal tolerance to eliminate patch redundancy.

Patch Processing

The patch images are L_2 normalised, shifted to zero mean, and scaled to unit standard deviation to approximately produce illumination invariance [115]. The patch processing procedure is provided as follows, given an image \mathbf{I} with resolution $R \times C = N$, where i and j identifies the row and column coordinates of an entire set of pixels \mathbb{I} . First, the image is normalised to unit length $\hat{I}_{ij} = (I_{ij}) / \|\mathbf{I}\|_2$, where

$$\|\mathbf{I}\|_2 = \left[\sum_{I_{i,j} \in \mathbb{I}} I_{i,j}^2 \right]^{1/2}. \quad (3.1)$$

then, the image is centralised $\bar{I}_{ij} = \hat{I}_{ij} - \hat{\mu}$, where

$$\hat{\mu} = \frac{\sum_{\hat{I}_{i,j} \in \hat{\mathbb{I}}} \hat{I}_{ij}}{N}. \quad (3.2)$$

Finally, the image is scaled to unit standard deviation by $\tilde{I}_{ij} = \bar{I}_{ij}/\bar{\sigma}$, where

$$\bar{\sigma} = \sqrt{\frac{\sum_{\bar{I}_{i,j} \in \bar{\mathbb{I}}} (\bar{\mu} - \bar{I}_{ij})^2}{N-1}}. \quad (3.3)$$

In this case, $\bar{\mu}$ is the mean value of $\bar{\mathbb{I}}$ and is zero since $\bar{\mathbb{I}}$ has already been centralised by $\hat{\mu}$ from Eq. 3.2. In practice, the above procedure can also be computed by reshaping the image patch into image vector by sequentially stacking all the columns of the image matrix into an one-dimensional (1D) column array. It should be noted that after the mean shift and unit standard deviation scaling, the image is no longer L_2 normalised. While the norm of the final image vector is not unity, it will always be $\sqrt{N-1}$.

Proof: $\|\tilde{x}\|_2 = \sqrt{\sum \tilde{x}_i^2} = \sqrt{\sum \bar{x}_i^2 (N-1) / \sum (\bar{\mu} - \bar{x}_i)^2} = \sqrt{\sum \bar{x}_i^2 (N-1) / \sum (-\bar{x}_i)^2} = \sqrt{N-1}$

3.1.2 Codebook Dictionary

Image spatial filter convolution has been used since the 1980s to mimic early stages of the primate visual system [113]. The filter kernels are models of simple cell receptive fields in the visual cortex. There are three categories of visual cells: cells with radially symmetric receptive fields which lead to DoG or LoG models; oriented odd-symmetric cells with receptive fields that can be modeled as rotated copies of a horizontal odd-symmetric receptive field; and oriented even-symmetric cells whose receptive fields can be modeled as rotated copies of a horizontal even-symmetric receptive field [113]. Schmid [117], Leung and Malik [116] have introduced sets of filter kernels to model the receptive fields of simple cells. The application of these filter banks to the local image patch allows the texture image to become scale and rotationally invariant and individual textures can be made distinguishable by different spatial averages of some locally computed neural response [111].

The Schmid (S) filter set consists of 13 rotationally invariant filters of the form

$$F(r, \sigma, \tau) = F_0(\sigma, \tau) + \cos\left(\frac{\pi\tau r}{\sigma}\right) \exp\left(-\frac{r^2}{2\sigma^2}\right), \quad (3.4)$$

where τ is the number of cycles of the harmonic function within the Gaussian envelope of the filter. $F_0(\sigma, \tau)$ is added to obtain a *DC-free* waveform, this makes the filter robust to illumination changes. The (σ, τ) pair takes the values of (2, 1), (4, 1), (4, 2), (6, 1), (6, 2), (6, 3), (8, 1), (8, 2), (8, 3), (10, 1), (10, 2), (10, 3), (10, 4).

The Maximum Response (MR) filter set consists of 38 filters but only 8 filter response maps are ultimately stored. The 38 filters are made from edge, bar, and spot filters [115]. The edge and bar filters each have 3 scales $(\sigma_x, \sigma_y) = \{(1, 3), (2, 6), (4, 12)\}$, and each has 6 orientations incremented by 30 degrees. In addition to the 36 edge and bar filters, the MR set also includes 2 isotropic filters that are the Gaussian and the LoG. From the 38 filter responses, only 8 filter responses are recorded by taking the maximum response of the anisotropic filters at each scale across all orientations [115]. The maximum response maps, three scales for two filters and two isotropic, from the incremental orientation filter to achieve rotational invariance.

Varma and Zisserman [115] have shown the S filter set and the MR8 filter set to be superior to other filter sets such as the Leung and Malik filter set, which consists of 48 filters with first and second derivatives of Gaussians at 6 orientations and 3 scales plus 8 LoG and 4 Gaussian isotropic filters. The MR filters provide good features for anisotropic textures, it overcomes the limitations of rotationally invariant filters which do not respond strongly to oriented image patches. Furthermore, computation cost increases significantly with the increasing number of filters. Keeping 8 maximum filter responses out of 38 dramatically reduces the computation time and allows correct rotational mapping of the texture. This investigation combines the Schmid and the MR8 filters to form a 21 kernel filter set (S+MR8) shown in Fig. 3.2. The filter bank elements are L_1 normalised to the same range. Each filter is 49 pixels in size. Once the patch images are filtered, they are contrast normalised using the following equation [115],

$$\hat{F}_{ij} = F_{ij} \frac{\log\left(1 + \frac{\|F(\mathbf{x})\|_2}{0.03}\right)}{\|F(\mathbf{x})\|_2}, \quad (3.5)$$

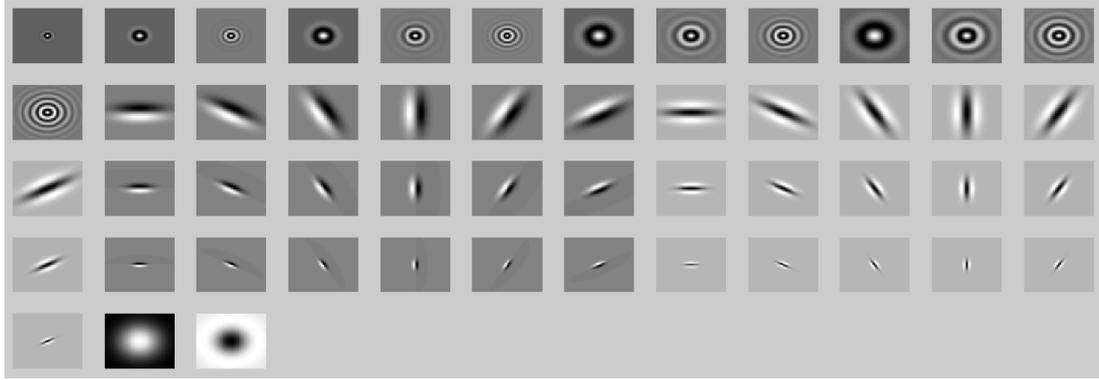


Figure 3.2: S+MR8 filter set with 13 S filters and 38 MR filters. Out of the 38 MR filters only 8 maximal response anisotropic maps are saved by collapsing the 6 orientations into a single response map.

where $F(x)$ is the filtered response at pixel x .

***k*-Means Clustering**

Analogous to the BoW text classification, BoVW requires a visual dictionary as vocabulary basis for constructing component models. Since image keypoints can form on multiple locations within the same spacecraft component, it is necessary to mean cluster similar filter response maps and combine redundant textons. The goal is to obtain a unique set of codevector vocabulary and develop normalised component model histograms. We apply PCA to *k*-Means for efficient cluster of filter responses into unique textons. Filter response map comparison is performed using Normalised Grayscale Correlation (NGC) similarity defined in Agarwal and Roth [354] and Leibe *et al.* [355] as,

$$NGC(p, q) = \frac{\sum_{i=1}^R \sum_{j=1}^C (p_{ij} - \mu_p)(q_{ij} - \mu_q)}{\sqrt{\sum_{i=1}^R \sum_{j=1}^C (p_{ij} - \mu_p)^2 \sum_{i=1}^R \sum_{j=1}^C (q_{ij} - \mu_q)^2}} = \hat{\mathbf{x}}^T \hat{\mathbf{y}}, \quad (3.6)$$

where $\hat{\mathbf{x}}$ and $\hat{\mathbf{y}}$ are centre-normalised image vectors of the p and q patches and the clustered similarity is

$$S(C_n, C_m) = \frac{\sum_{p \in C_n, q \in C_m} NGC(p, q)}{|C_n||C_m|} > tol, \quad (3.7)$$

where C_n and C_m are the number of patches in clusters \mathbb{C}_n and \mathbb{C}_m respectively, and tol is the similarity threshold for decide whether to combine the clusters. For a single patch, the cluster recursive averaging is

$$\bar{\mathbf{x}}_{n+1} = \frac{C_n \bar{\mathbf{x}}_n + \hat{\mathbf{x}}}{C_n + 1}, \quad (3.8)$$

In general, the recursive average of two clusters is

$$\bar{\mathbf{x}}_{n+m} = \frac{C_n \bar{\mathbf{x}}_n + C_m \bar{\mathbf{x}}_m}{C_n + C_m}. \quad (3.9)$$

Hence, Eq. (3.7) can be simplified to

$$S_{n,m} = \bar{\mathbf{x}}_n^T \bar{\mathbf{x}}_m. \quad (3.10)$$

An example of image vector clustering is shown in Fig. 3.3. Each vertical bar represents a filter response map image vector. The first layer represents the input and the second and third layers show clustering within the image local regions. The final layer shows clustering of global mean image vectors. Our first round of clustering is performed within local groups defined by a local region bin and not by exhaustive search across all image patches; this dramatically reduces the processing time and has little influence on the overall results since the global mean image vectors are clustered in the final step. Figure 3.4 shows the clustering process for a solar panel. In this example, the textons are formed from the normalised image patches. The patches that are clustered into the final textons have the same colour border as the three final textons. Additional solar panel images and spacecraft components will result in more textons in the codebook. We select the solar panel, station truss, and the ISS module as our test components. For each vehicle component, three representative images are selected for developing texton code vectors. Each training image sets have different viewpoint, illumination, and zoom. Figure 3.5 shows the S-MR8 filtered results of the solar panel example. We cluster the individual component images, and later over all the components for a unique texton set.

Patches sharing the same local region is compared to reduce the number of k -Means similarity. We first determine the minimum and maximum bounding keypoints and generate an integer map base on these two diagonally opposite corners. We then loops through all the keypoints stamping the integer map with a local region bin number as per Fig. 3.6,

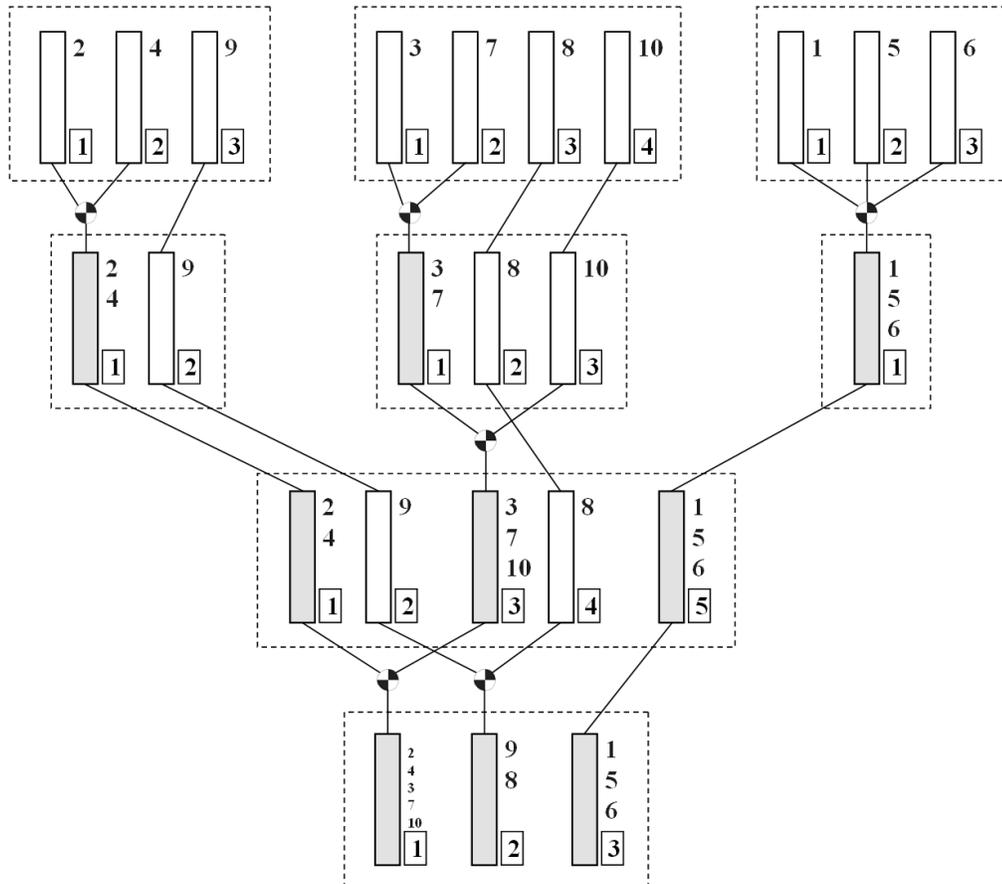


Figure 3.3: Image patch clustering. Blank columns represent the original image patches and shaded columns represent clustered mean images. Keypoint ID of each image patch is stored with each clustered group.

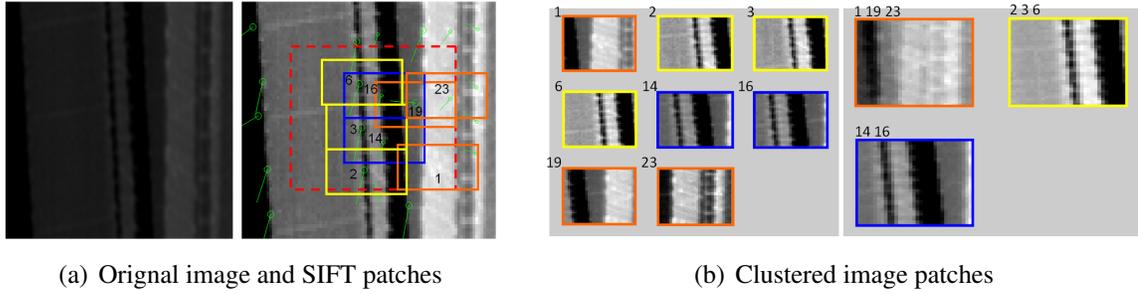


Figure 3.4: Solar panel image texton example. The solar panel image is processed for SIFT keypoints. Only the keypoints inside the boundary (red broken lines) are used. Image patches around the keypoints are extracted. Image patches are clustered using the k -Means method. Index 1 – 19 – 23 forms the first texton, 2 – 3 – 6 forms the second texton, and 14 – 16 forms the third texton.

if the next keypoint is on top of a stamped area, then that keypoint is assigned the bin number underneath, or else the keypoint is assigned 0 and a new bin number is issued. This method groups the keypoints into local region bins so the closest patches are clustered first. After local patches are clustered by NGC similarity, we cluster one last time to obtain global means. This local region scheme reduces the order of the NGC checks from $\mathcal{O}(M^2)$ to $\mathcal{O}(Mm + 1)$, where M is the number of total keypoints and m is a smaller subset of keypoints sharing the same bin.

For faster assignment of the query patch images to the codebook textons, only the primary eigen components of the textons images are stored using PCA. Details of PCA image transform and comparison may be found in Chapter 5. Once transformed into the eigenspace, the transformation matrix, $\tilde{\mathbf{P}}_{K \times N}^T$, and the eigenspace texton database, $\tilde{\mathbf{G}}_{K \times M}$, are stored for the inference phase. PCA can also be used to reduce the dimension of the component model feature vector defined in Sec. 3.1.2. Since the codebook dictionary can be a large number of textons, the classification process normally requires computing the inverse of the feature space covariance which can run into computation resource and timing issues. The application of PCA in this case will minimise both challenges.

The number of textons in our ISS example can range between 595 to 5495 for similarity threshold of 0.75 to no clustering respectively. The numerous textons can be time-consuming to compare with during the inference phase. One way of reducing the number of textons is by random selection. A more logical approach is to eliminate the textons that are shared across multiple components. The shared textons represent common ‘words’

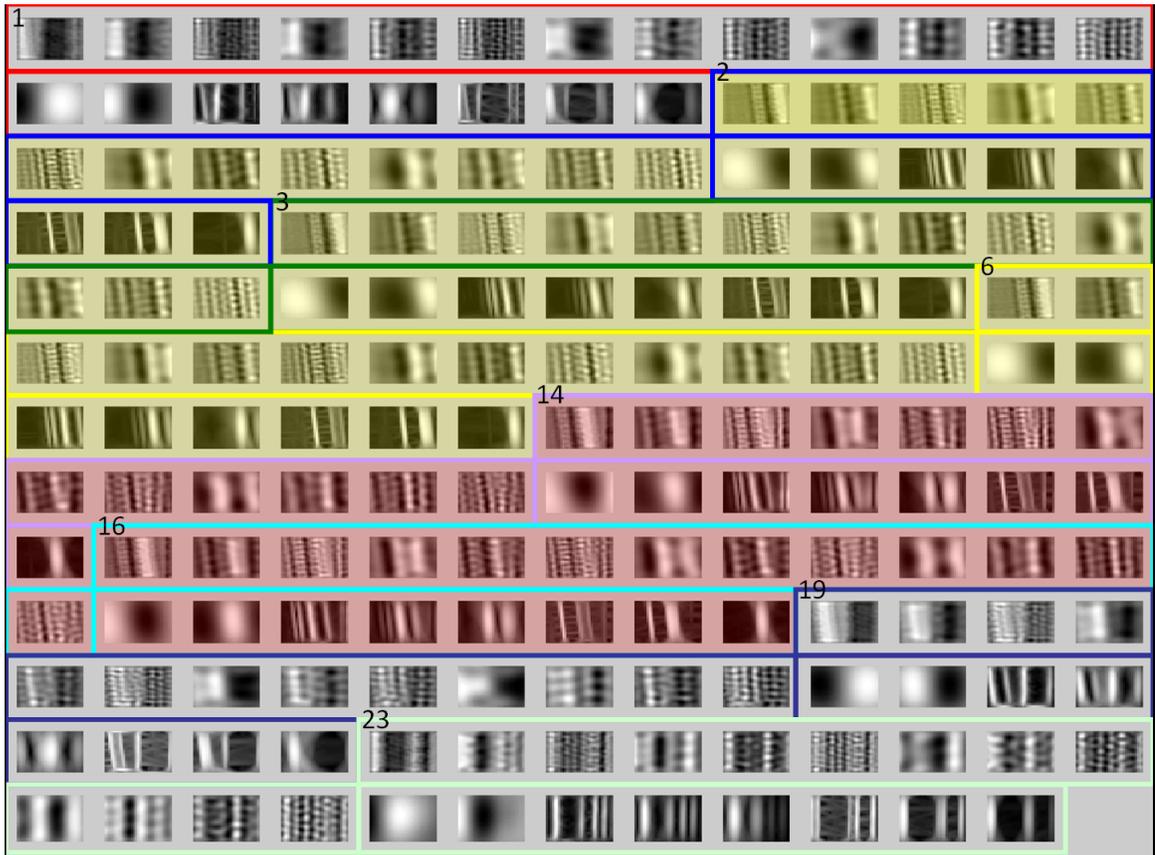


Figure 3.5: S-MR8 filtered response maps for solar panel keypoint patches. Filter response map with the same colour shading belongs to the same image group from Fig. 3.4.

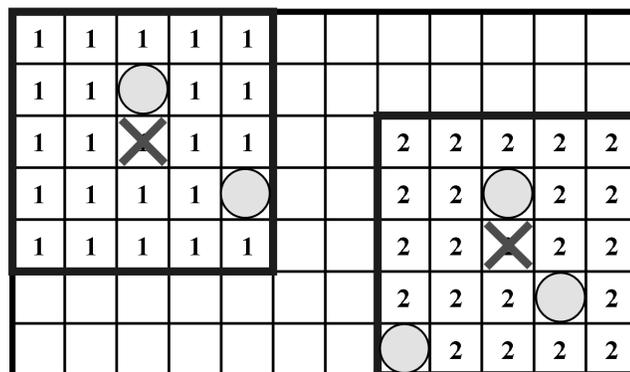


Figure 3.6: Local region bins; crosses represents the central keypoint, shaded circles represents keypoints that are assigned to the local region bin.

amongst component materials which could lead to the feature space to be less separable and more erroneous to classify. The codebook reduction approach used in our example deletes the texton that has multiple image associations, this can lead to a 20 percent decrease to the overall codebook. Future investigations may examine more surgical removal of texton between components while keeping the non-image-unique texton within the same component.

Object Vector Representation

The labeled training images are filtered and grouped into component categories. The extracted codevectors are organised into a codebook visual vocabulary. Next, feature vectors for each component may be generated based on its codevector occurrence frequency. The feature vector can also be viewed as a histogram model; it is equivalent to the ‘keyword phase’ and ‘search sentence’ in the BoW analogy. To ensure comparability, each feature vector is normalised to the unit length. A stack of the feature vectors are created for all the identified training component classes. Multiple feature vectors can be formed to represent different images of the same component class under illumination, zoom, rotation, and viewpoint changes. Figure 3.7 shows a simple example of textons forming the codebook and the formation of two normalised feature vectors storing the occurrence frequency of the basis codevectors. The same procedure of object feature vector generation is carried out on the inference image.

3.1.3 Classifiers

The designation of the object feature vectors can be performed using linear, non-linear, or non-parametric classifiers. A rich set of feature vectors will increase classifier reliability during inference. The linear classifiers evaluated are the Fisher’s Discriminant (FD) and the Ho-Kashyap (HK) method; the non-linear classifier evaluated is the Bayes Optimal (BO) discriminant in quadratic form; finally, two non-parametric classifiers evaluated are the k -Nearest Neighbour (k NN) and the Naïve Bayes (NB) Classifier. While linear classifiers are simple to form, their precision depends on the separability of the feature vectors between components. For components that have a similar texture, non-parametric classifiers should, in theory, have higher performance. In addition to the evaluated classifiers, future work

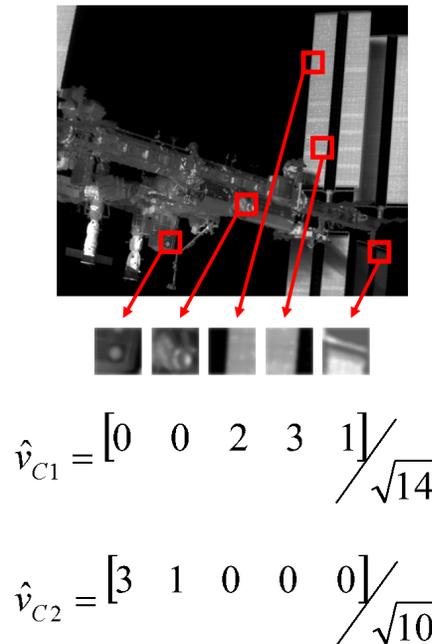


Figure 3.7: Example of a codebook forming an object feature vector.

may also examine Support Vector Machine [121] and the χ^2 test [140]. Convolutional neural networks such as YOLO [135] are also powerful bounding box classification tools. The ConvNet bounding box techniques will be discussed in detail in Sec. 3.2.

2-class to 3-class Classification

Methods presented in this section are defined for two-class classifiers. The two-class classifier can be extended to multi-class by computing all combinations of the class discrimination functions. During inference, the query is decided between the first and second classes, the resulting class is compared with the third class, then the next result is compared with the fourth class and so forth. This cascading classification method is simple to implement and is scalable to higher class orders. More sophisticated generic multi-class discrimination techniques [356] are also available and may be evaluated in the future. Specifically for the three-class system, the classification method is presented in algorithm 1.

Algorithm 1 THREE_CLASS_CLASSIFIER

```

1: procedure THREE_CLASS_CLASSIFIER( $\mathbf{x}, \mathbf{A}_i, \mathbf{B}_i, C_i$ )
2:   for  $i = 1, i++,$  while  $i \leq 3$  do
3:      $\mathcal{F}_i \leftarrow \mathbf{x}^T \mathbf{A}_i \mathbf{x} + \mathbf{B}_i^T \mathbf{x} + C_i$ 
4:   if  $\mathcal{F}_1 > 0$  then
5:     if  $\mathcal{F}_3 > 0$  then
6:        $x\_class\_label \leftarrow 3$ 
7:     else
8:        $x\_class\_label \leftarrow 1$ 
9:   else
10:    if  $\mathcal{F}_2 > 0$  then
11:       $x\_class\_label \leftarrow 2$ 
12:    else
13:       $x\_class\_label \leftarrow 3$ 

```

Fisher's Discriminant

The Fisher's Discriminant (FD) [356] function is a linear classification decision boundary; this method collapses the feature space on to an 1D line thereby minimises the so-called *curse of dimensionality*. Given a feature vector \mathbf{x}_{ij} of the length N textons, it is the j^{th} image out of M_i local component images and i^{th} class out of C classes. The set of textons in the codebook, local class component model images, and object classes are denoted by \mathbb{N} , \mathbb{M}_i , and \mathbb{C} respectively. There are M feature vectors in the global set of model images, denoted by \mathbb{M} . The feature vector is collapsed on to a single dimension by $y_{ij} = \mathbf{w}^T \mathbf{x}_{ij}$, where \mathbf{w} is the 1D line direction. The objective is to compute the weights in \mathbf{w} to produce maximum separability after the projection. Let define $\boldsymbol{\mu}_i$ as the mean of \mathbf{x}_{ij} over all the local component images, then $\tilde{m}_i = \mathbf{w}^T \boldsymbol{\mu}_i$ is the 1D projected mean. The FD cost function is defined as,

$$J(\mathbf{w}) = \frac{\mathbf{w}^T \mathbf{S}_B \mathbf{w}}{\mathbf{w}^T \mathbf{S}_w \mathbf{w}}. \quad (3.11)$$

Equation 3.11 is known as the *Rayleigh Quotient*, where \mathbf{S}_B is the *between-class scatter matrix* $\mathbf{S}_B = (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^T$, and \mathbf{S}_w is the *within-class scatter matrix*

$$\mathbf{S}_w = \sum_{\mathbf{x}_{ij}, \boldsymbol{\mu}_i \in \mathbb{C}} \sum_{\mathbf{x}_{ij} \in \mathbb{M}_i} (\mathbf{x}_{ij} - \boldsymbol{\mu}_i)(\mathbf{x}_{ij} - \boldsymbol{\mu}_i)^T. \quad (3.12)$$

The numerator from Eq. 3.11 is the square mean difference between the two classes $(\tilde{m}_1 - \tilde{m}_2)^2$, the denominator is the sum of the scatter over all class i . The scatter over a single class can be computed as $\tilde{S}_i^2 = \sum_{y_{ij} \in \mathbb{M}_i} (y_{ij} - \tilde{m}_i)^2$. For optimal discrimination, the *between-class* scatter is maximised and *within-class* scatter is minimised by letting the partial derivative of the cost with respect to the weights to be zero. The decision boundary or *canonical variate* can be derived as

$$\mathbf{w} = \mathbf{S}_w^{-1} (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2). \quad (3.13)$$

The FD computation is on the order of $\mathcal{O}(N^2M)$. In practice, feature vector with the length of the full codebook results in a *within-class scatter matrix* that is too large to solve for its inverse, to reduce the sample dimension, PCA is applied prior to the FD operation to reduce the feature dimension. The PCA process also reduces the computation order from N to the number of principal components. During inference, the query feature vector is tested using the FD decision rule:

$$\mathbf{w}^T (\mathbf{x} - \boldsymbol{\mu}) \underset{\omega_2}{\overset{\omega_1}{\geq}} 0, \quad (3.14)$$

where $\boldsymbol{\mu}$ is the mean of feature vectors over all local component images and classes.

Ho-Kashyap Procedure

The Ho-Kashyap (HK) [356] linear discrimination function is similar to the *perceptron* and is based on the numerical method of gradient descent. Given the general formulation for the linear discrimination function $\mathcal{F} = \mathbf{w}^T \mathbf{x} + w_0$, it can be reformulated as $\mathbf{Y} \mathbf{a} = \mathbf{b}$ where

$$\mathbf{Y} = \begin{bmatrix} \mathbf{1}_1 & \mathbf{X}_1 \\ -\mathbf{1}_2 & -\mathbf{X}_2 \end{bmatrix}, \quad (3.15)$$

and \mathbf{X}_i is the stacked transpose of training feature vectors for class i . The decision weights are grouped into $\mathbf{a}^T = \begin{bmatrix} w_0 & \mathbf{w}^T \end{bmatrix}$. For linearly separable samples, $\mathbf{b} > 0$. The cost

criterion for the HK procedure is

$$J(\mathbf{a}, \mathbf{b}) = \|\mathbf{Y}\mathbf{a} - \mathbf{b}\|_2^2. \quad (3.16)$$

Discrimination line with maximum separability is found by minimizing the cost criterion; this is done by taking the gradient of J with respect to \mathbf{a} and \mathbf{b} . Unlike FD, HK does not require the mean estimate in its calculation. The decision boundary weights in \mathbf{a} are computed using the *minimum-squared-error pseudoinverse* (\mathbf{Y}^\dagger) of \mathbf{Y} ,

$$\mathbf{a} = \left(\mathbf{Y}^T \mathbf{Y}\right)^{-1} \mathbf{Y}^T \mathbf{b} = \mathbf{Y}^\dagger \mathbf{b}. \quad (3.17)$$

Consequently, the entire training data set is used in generating the discrimination function and Eq. 3.17 allows the cost criterion with respect to \mathbf{a} to go to zero. \mathbf{b} on the other hand is computed using an optimal descent based on the \mathbf{b} gradient of the cost criterion,

$$\mathbf{b}(k+1) = \mathbf{b}(k) + 2\eta \mathbf{e}^+(k), \quad (3.18)$$

where η is the stepsize, k is the step index, and

$$\mathbf{e}^+(k) = \frac{1}{2}(\mathbf{e} + |\mathbf{e}|). \quad (3.19)$$

The \mathbf{e} value is computed by,

$$\mathbf{e}(k) = \mathbf{Y}\mathbf{a}(k) - \mathbf{b}(k), \quad (3.20)$$

where iteration is completed when \mathbf{e} is less than some minimum threshold (set to 10^{-14} in this investigation). We designate the aforementioned HK algorithm as HK-0, and explore two additional HK variations. The first variation, HK-1, doubles \mathbf{e}^+ ; whereas the second variation, HK-2, approximate \mathbf{a} using a second gradient descent,

$$\mathbf{a}(k+1) = \mathbf{a}(k) + \eta \mathbf{R} \mathbf{Y}^T |\mathbf{e}(k)|, \quad (3.21)$$

where \mathbf{R} is an arbitrary, constant, positive-definite matrix that can be optimised as,

$$\mathbf{R} = \frac{1}{\eta} \left(\mathbf{Y}^T \mathbf{Y}\right)^{-1}. \quad (3.22)$$

Bayes Optimal Classifier

The Bayes Optimal (BO) quadratic form discrimination approach implements the Bayes decision rule. It assumes Gaussian normal distribution for the probability density function and provides an statistical optimal division of the component classes. The BO classifier is rooted in the *Bayes formula* where the *posterior* is the *likelihood* times the *prior* over the *evidence*. Specifically, the *posterior* is the probability of component class given the feature vector $P(\omega_i|\mathbf{x})$, the *likelihood* is the probability of the feature vector given the component class $P(\mathbf{x}|\omega_i)$, the *prior* is the probability of the component class $P(\omega_i)$, and the *evidence* is the total probability of the feature vector $P(\mathbf{x})$, where ω_i represent the i^{th} class set out of \mathbb{C} classes. The most likely component class can be determined by observing the query feature vector,

$$P(\omega_i|\mathbf{x}) = \frac{P(\mathbf{x}|\omega_i)P(\omega_i)}{P(\mathbf{x})}. \quad (3.23)$$

The Bayes decision rule can be formally established as a class comparison of the *posterior* or *a posteriori*.

$$P(\omega_1|\mathbf{x}) \underset{\omega_2}{\overset{\omega_1}{\geq}} P(\omega_2|\mathbf{x}) \quad (3.24)$$

Substituting Eq. 3.23 into Eq. 3.24, the *evidence* in the denominator can be cancelled. Using the multi-dimensional Gaussian normal distribution to approximate the training set *likelihood*, the quadratic form of the optimal Bayes formulation can be derived as

$$\mathbf{x}^T \mathbf{A} \mathbf{x} + \mathbf{B}^T \mathbf{x} + C \underset{\omega_2}{\overset{\omega_1}{\geq}} 0, \quad (3.25)$$

where

$$\mathbf{A} = \Sigma_2^{-1} - \Sigma_1^{-1}, \quad (3.26)$$

$$\mathbf{B}^T = 2 \left[\boldsymbol{\mu}_1^T \Sigma_1^{-1} - \boldsymbol{\mu}_2^T \Sigma_2^{-1} \right], \quad (3.27)$$

and

$$C = \left(\boldsymbol{\mu}_2^T \Sigma_2^{-1} \boldsymbol{\mu}_2 - \boldsymbol{\mu}_1^T \Sigma_1^{-1} \boldsymbol{\mu}_1 \right) - 2 \left(\ln \frac{P_2}{P_1} + \frac{1}{2} \ln \left(\frac{|\Sigma_1|}{|\Sigma_2|} \right) \right). \quad (3.28)$$

P_i is the previously defined *prior* or *apriori* probability, $\boldsymbol{\mu}_i$ and $\boldsymbol{\Sigma}_i$ are the mean and covariance matrices from the training data set respectively. For equal number of training images in the individual classes, the *apriori* ratio term goes to zero or else the *apriori* is $P_i = M_i/M$ where M is defined in Sec. 3.1.3. The *unbiased sample covariance matrix* is computed as

$$\boldsymbol{\Sigma}_i = \frac{1}{M_i - 1} \sum_{\mathbf{x}_{ij} \in \mathbb{M}_i} (\mathbf{x}_{ij} - \boldsymbol{\mu}_i)(\mathbf{x}_{ij} - \boldsymbol{\mu}_i)^T. \quad (3.29)$$

PCA is applied to the feature space prior to the execution of the BO procedure for manageable covariance inverse calculation.

***k*-Nearest Neighbour**

A simple approach to determine the component class of a query feature vector is to adopt the label of majority of the k nearest training data neighbour; this is the so-called k -Nearest Neighbour (k NN) method. There are many ways to measure the neighbour distance, this includes the use of Minkowski distance (p -norms), where p can be $-\infty$ to ∞ using

$$\bar{r}_{ij} = \left(\sum_{x_{ijl} \in \mathbb{N}} (x_l - x_{ijl})^p \right)^{1/p}, \quad (3.30)$$

where x_l is the l^{th} normalised texton count in the query feature vector. \mathbb{N} is the set of all the textons in the codebook dictionary, there are a total of N textons in this set. Alternatively, using *cosine similarity* as the nearest neighbour metric collapse the problem into a single dimension. *cosine similarity* is computed by the inner dot-product between the query feature vector and the training data as follows,

$$\theta_{ij} = \arccos \left(\frac{\mathbf{x} \cdot \mathbf{x}_{ij}}{\|\mathbf{x}\|_2 \|\mathbf{x}_{ij}\|_2} \right). \quad (3.31)$$

We evaluate the Hamming distance 0-norm (k NN-0), *i.e.* $\sum_{l=1}^N (|x_l - x_{ijl}| > 0)$; Manhattan 1-norm (k NN-1), *i.e.* $\sum_{l=1}^N |x_l - x_{ijl}|$; Euclidian 2-norm (k NN-2); ∞ -norm (k NN-i), *i.e.* $\max_l |x_l - x_{ijl}|$; $-\infty$ -norm (k NN-n), *i.e.* $\min_l |x_l - x_{ijl}|$; and the cosine similarity (k NN- θ). k NN does not require pre-inference training calculation, but requires training data during inference. The main drawback of the k NN method is its low speed when the training data

set is large due to full permutation of the distance measure. Future developments may investigate into more efficient methods such as *k-dimensional tree* [357], locality-sensitive hashing [358], and inverted multi-index [359, 360].

Naïve Bayes Classifier

The *multinomial* Naïve Bayes (NB) [361] model is a probabilistic supervised learning method for data classification. Adapting NB for BoVW, the ‘naïvety’ in NB assumes textons from the codebook are independent when forming the image model feature vector. Let us define a query feature vector \mathbf{x} composed by the subset \mathbb{D} of D textons from the codebook set \mathbb{N} . Let $\tilde{\mathbf{x}}$ be the non-normalised version of \mathbf{x} containing the frequency of occurrence for each texton such that $\tilde{\mathbf{x}}^T = [T_1 \ \dots \ T_N]$. For a single keypoint, the non-normalised feature vector is a $N \times 1$ array containing 1s and 0s, where $D = \sum_{x_l \in \mathbb{N}} T_l$. The probability of the query feature vector given the i^{th} component class $P(\tilde{\mathbf{x}}|\omega_i)$ is the product of all probability of the individual query textons given the i^{th} component class $P(\tilde{x}_l|\omega_i)$, the numerator in the *Bayes formula* given by Eq. 3.23 becomes

$$P(\omega_i|\mathbf{x}) \propto P(\omega_i) \prod_{\tilde{x}_l \in \mathbb{D}} P(\tilde{x}_l|\omega_i). \quad (3.32)$$

The objective is to find the component class with the maximum likelihood given the query keypoint’s feature vector. Using the logarithmic property of $\log(xy) = \log(x) + \log(y)$, the NB maximum likelihood component class, also known as the *maximum a posteriori* (c_{map}), can be derived as

$$\begin{aligned} c_{map} &= \operatorname{argmax}_{\omega_i \in \mathbb{C}} \tilde{P}(\omega_i|\tilde{\mathbf{x}}) \\ &= \operatorname{argmax}_{\omega_i \in \mathbb{C}} \left(P(\omega_i) \prod_{\tilde{x}_l \in \mathbb{D}} P(\tilde{x}_l|\omega_i) \right) \\ &= \operatorname{argmax}_{\omega_i \in \mathbb{C}} \left(\log P(\omega_i) + \sum_{l=1}^N T_l \log \tilde{P}(\tilde{x}_l|\omega_i) \right), \end{aligned} \quad (3.33)$$

where the *a priori* is computed the same way as in the BO Classifier. Note $\tilde{P}(\omega_i|\tilde{\mathbf{x}})$ is not the probability of the component class given the feature vector since the calculation of $P(\tilde{\mathbf{x}})$ from Eq. 3.23 is omitted, *i.e.* the *posterior* will not be out of one. This absolute

probability, however, have little significance because the interest is to find the most likely component class. In practice, the actual $\tilde{P}(\omega_i|\tilde{\mathbf{x}})$ is not known and can only be approximated by the training data. The accuracy of the *a posteriori* is subject to the volume and quality of the training sample in relation to the query images. Finally, if no textons are counted for a given class, zero $P(\tilde{x}_l|\omega_i)$ will result in a numerically inadmissible output from the logarithmic calculation. To elevate this challenge, *Laplace smoothing* is applied to the $P(\tilde{x}_l|\omega_i)$ calculation (denoted by $\tilde{P}(\tilde{x}_l|\omega_i)$) as follows,

$$\tilde{P}(\tilde{x}_l|\omega_i) = \frac{\sum_{j=1}^{M_i} T_{ijl} + 1}{\sum_{l=1}^N \sum_{j=1}^{M_i} T_{ijl} + N}. \quad (3.34)$$

Where T_{ijl} is the number of l^{th} textons counted for the j^{th} local component image and i^{th} component class from the training data, and $\sum_{l=1}^N \sum_{j=1}^{M_i} T_{ijl}$ is the total textons counted for the i^{th} component class. Equation 3.33 can be extended to use multiple keypoints in defining a query feature vector. In such case, the \tilde{x}_l is no longer binary but will be the frequency of the l^{th} texton occurrence over all the selected keypoints, and multiple keypoints will result in a stronger probable outcome of the component class.

3.1.4 Training and Inference

ISS hardware component training images are extracted from a set of 36 representative thermal images taken by the Neptec *TriDAR* IR camera during the NASA SSO missions STS-128 and STS-131. Different segments of the same STS flights are used in our evaluation. Ground truth bounding box was manually computed on the ISS thermal images. Individual component training images used to extract the texton codebook are shown in Fig. 3.8, 3.9, and 3.10 for station module, solar panel, and ISS truss segment respectively. The Jaccard criteria for bounding box performance was used to evaluate algorithm performance. The *Jaccard index*, also known as *Intersect over Union* (IoU), measures the predicted bounding box accuracy over the ground truth. The IoU for the i^{th} class is defined as

$$J_i = \frac{|x_i \cap y_i|}{|x_i \cup y_i|}, \quad (3.35)$$

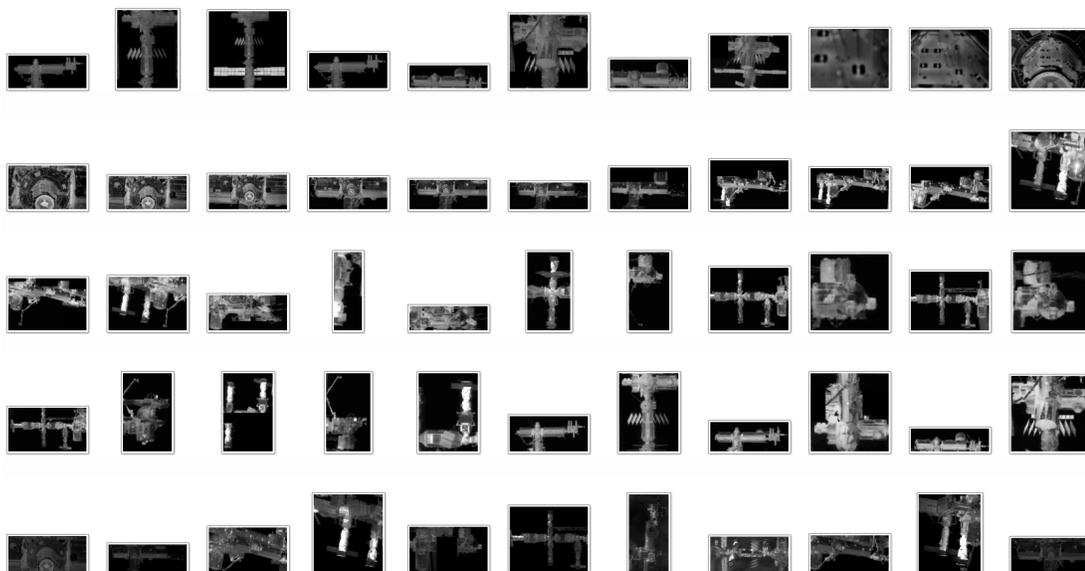


Figure 3.8: ISS station module training images used for component recognition training.

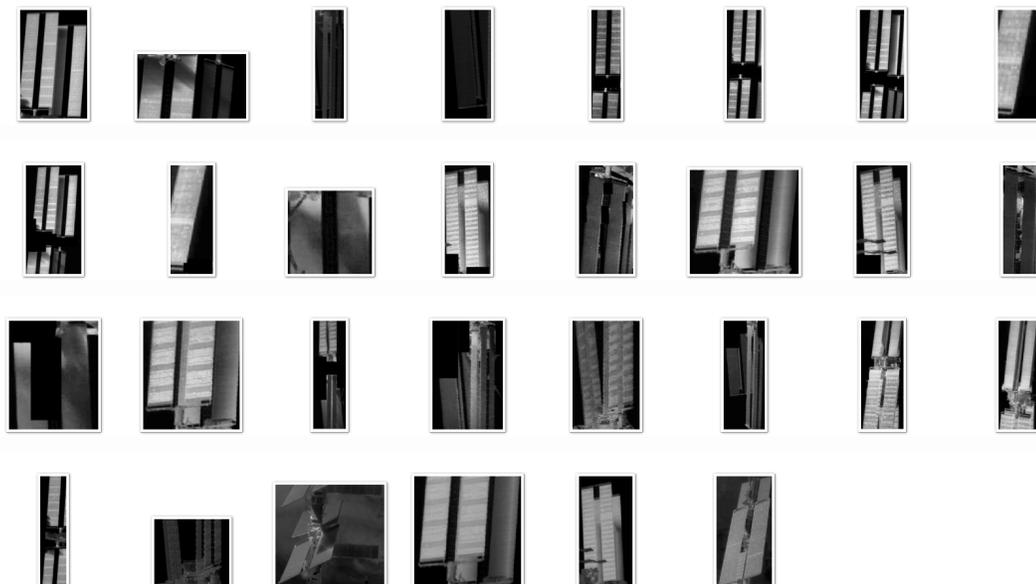


Figure 3.9: ISS solar panel training images used for component recognition training.

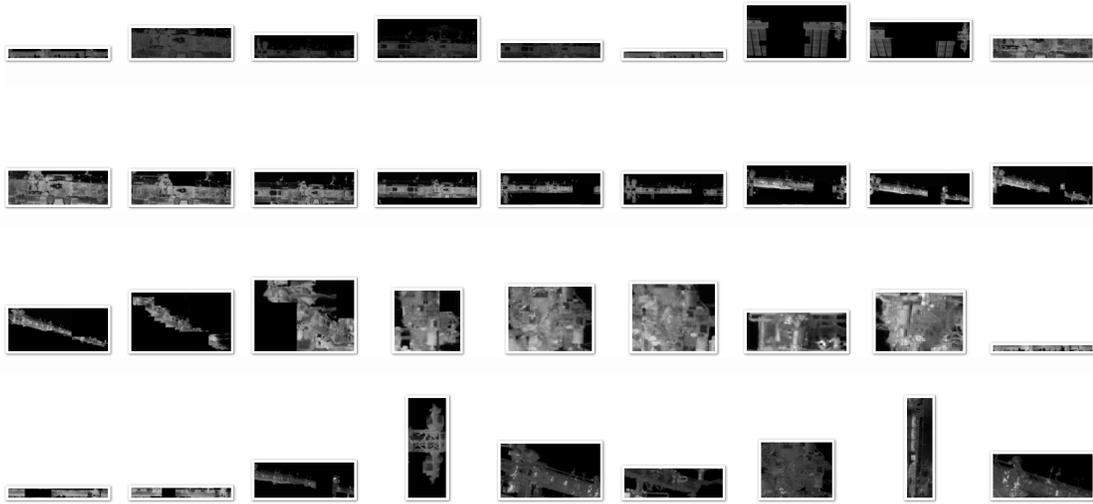


Figure 3.10: ISS truss segment training images used for component recognition training.

where x is the ground truth bounding box area and y is the predicted bounding box area.

The Receiver Operating Characteristics (ROC) [351] is a standard tool for measuring data retrieval. Analogous to document retrieval, the predicted and ground truth bounding box is related to the retrieved and ground truth documents respectively. The ROC precision, also known as the Positive Predicted Value (PPV) and is defined as

$$Precision = \frac{AP \cap EP}{EP} = \frac{TP}{EP}, \quad (3.36)$$

where AP is the actual positive or actual matches, TP is the true positive or true match. The True match is defined as correct overlay of the bound box area. EP is the estimated positive or total query matches, it is the sum of TP and the false positive (FP). The false negative (FN) can be computed as the AP minus TP . The ROC recall is defined as the true matches over the actual matches as follows

$$Recall = \frac{AP \cap EP}{AP} = \frac{TP}{AP}, \quad (3.37)$$

The ROC accuracy is defined as

$$Acry_i = \frac{(TP_i + TN_i)}{n_p}, \quad (3.38)$$

where n_p is the entire sample space. The Jaccard criteria from Eq. 3.39 can be defined in terms of ROC. A single class IoU is equivalent to

$$J_i = \frac{TP_i}{(EP_i + FN_i)}. \quad (3.39)$$

The total IoU score for each component class is the mean IoU scores over all computed frames.

3.1.5 Localisation Performance

A parameter hyperspace can be formed to evaluate the performance of the BoVW. The most sensitive factors in the bounding box precision are the query image keypoint response map to texton similarity tolerance; the number of keypoints used in the query image bounding box; the PCA compression order for textons and feature vectors; and the type of classifier and training image size. This section examines the effects of each hyperparameter.

Texton Similarity

Once the filter response map is computed for the query image, it must be labeled with the texton designation. Texton label is assigned to the most similar patch image by using Eq. 3.7. The final keypoint classification precision is sensitive to the similarity tolerance used for texton assignment. Since texton image vectors are not normalised to one, the similarity tolerance can take on values greater than one. The results show BO was not affected by the similarity tolerance between 10 to 20, and optimal tolerance of 15 was selected based on keypoint classification precision. Results of the optimal similarity tolerance is shown in Table 3.1.

Query Keypoints

A component bounding box is computed using two most outer keypoints. Practically, the keypoints are too numerous to be fully used in real-time. For example, a single 640×480 image can produce nearly a thousand keypoints as shown in Fig. 3.11; this shows the keypoints are a reliable way of mapping the spacecraft object from its surroundings. Bounding

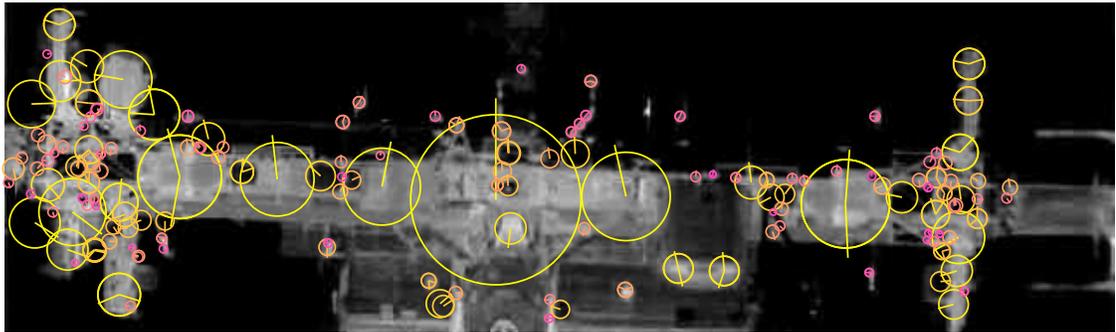
Table 3.1: Comparison of keypoint precision using different classification techniques and texton label similarity tolerance. Results are computed based on an ISS thermal camera image, keypoint classification is performed manually. Hyper parameters are set as follows: 50 keypoints; no texton image PCA; feature vectors used in FD, HK-0, BO, and k NN-2, are converted to 10 principal component eigenspace. No PCA was applied to the NB method. The unique codebook of 595 textons was used as BoVW basis.

Similarity Tolerance	10	12	15	17	20
FD	10	67	76	81	81
HK-0	14	67	81	81	81
BO	81	81	81	81	81
k NN-2	2	74	52	33	17
NB	52	38	21	5	2

box precision is proportional to the number of keypoints selected; however, it is also proportional to the computation costs. To remove excessive keypoints, NMS is applied to a region of 64 pixel-radius from the keypoint center and are evenly spread across the image; this allows each keypoint to occupy roughly 10 percent of the image region or approximately 100 keypoints per image. The query keypoints are reduced further by random selection. Future investigations may use keypoint strength instead of random selection. The number of keypoints used in our investigation is 50 per Table 3.3.

Table 3.2: Comparison of keypoint precision using various classification techniques and number of keypoints. Hyper parameters: similarity tolerance is 15; no texton image PCA; feature vectors used in FD, HK-0, BO, and k NN-2, are converted to 10 principal component eigenspace. No PCA was applied to the NB method. The unique codebook of 595 textons was used as BoVW basis. Computation performed on Intel® Core™ 2 Quad Q6600 – 2.4GHz Processor running on 32bit Windows-Vista-SP2. MATLAB 7.0 R14 was used for coding.

Keypoints	10	20	30	40	50	100
Query per frame	19	29	48	58	61	107
Time (s)						
FD	40	67	76	81	81	81
HK-0	40	67	81	81	81	81
BO	40	81	81	81	81	81
k NN-2	40	74	52	33	17	17
NB	40	38	21	5	2	2



(a) SIFT Keypoints



(b) Reduced Patches

Figure 3.11: SIFT keypoint and patches on the ISS infrared image. The keypoints in (a) is before NMS and random keypoint selection. The patches in (b) is after NMS and random keypoint selection reducing the number of keypoints to 30.

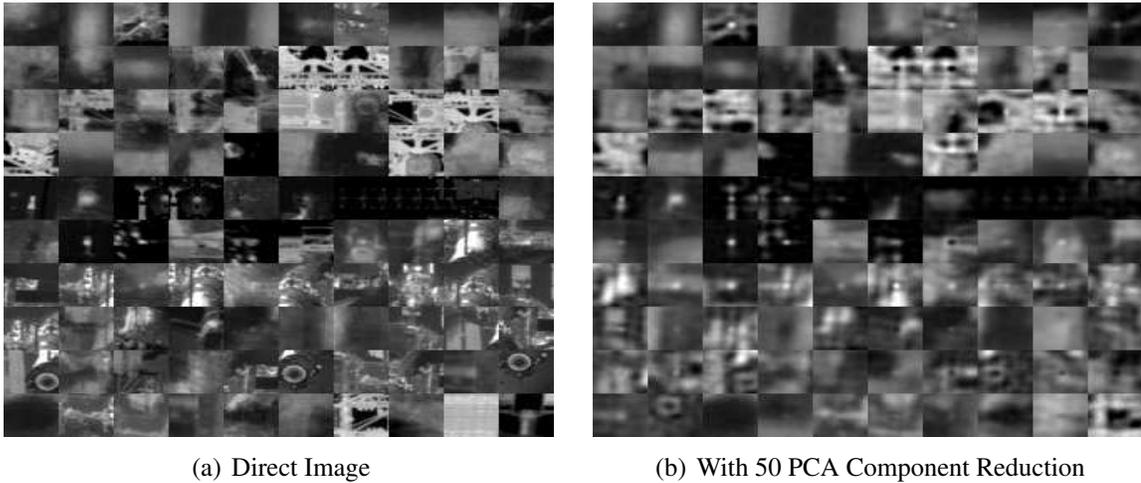


Figure 3.12: Unfiltered image patches with 32 pixel sides. 50 principal components were kept and converted back to the image domain.

Texton and Feature Vector PCA

We used PCA for texton and feature vector compression. Texton PCA compression increases image blur when removing a higher number of principal components, some examples of the patch compression is provided in Fig. 3.12. We tested 0, 3, 5, 10, 20, 50, 100, 200 and 500 principal components. Experiments show texton PCA compression did not cause significant change to the final keypoint classification or significantly impacted classification time; this is mostly because the size of the texton images is 10 pixels in height and width. The small response map with rudimentary features such as edges and blobs are less influenced by PCA compression blurring. While texton PCA is optional, feature vector PCA is necessary for parametric classifiers. The evaluated parametric methods all have some matrix inversion. The matrix inversions are only computationally feasible if PCA can reduce the matrix dimension.

Training Data and Classifier Performance

We evaluated five classifiers for spacecraft bounding box generation, Table 3.3 provides the precision results. HK-0 and 1 outperforms HK-2, and k NN-2 is a robust classifier in comparison to the other k NN methods. The 740 texton codebook is non-unique and performed worse than the unique codebook. Table 3.4 provides the mean Jaccard index for

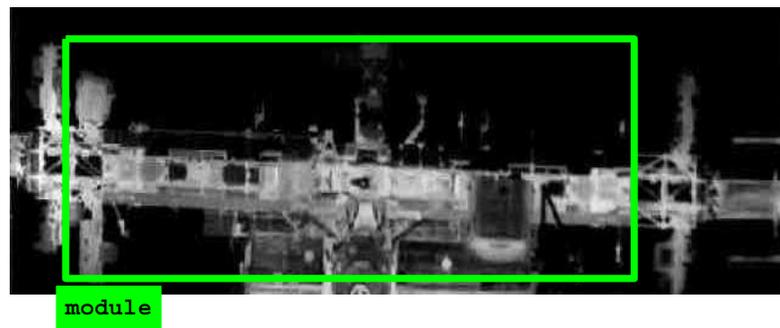
the different classification methods. The parametric methods has slightly outperformed the non-parametric ones. Figure 3.13 shows qualitative examples of ISS component detection using textons. Figure 3.13(a) shows incorrect prediction of the module where portion of the truss structure on the left side of the image was included in the bounding box. Figure 3.13(b) shows a correct bounding box prediction for the ISS truss. In summary, the BoVW method can perform object location, however, it does not have the desired precision for target extraction. In the next section, we evaluate the latest deep learning approach with higher performance than the traditional machine learning recognition and localisation.

Table 3.3: Comparison of keypoint precision using different classification techniques and codebook size. Hyper paramters: similarity tolerance is 15; 50 keypoints; no texton image PCA; feature vectors used in FD, HK-0 – 2, BO, and kNN -0 – 2, ∞ , $inf\theta$, θ , are converted to 10 principal component eigenspace. No PCA was applied to the NB method.

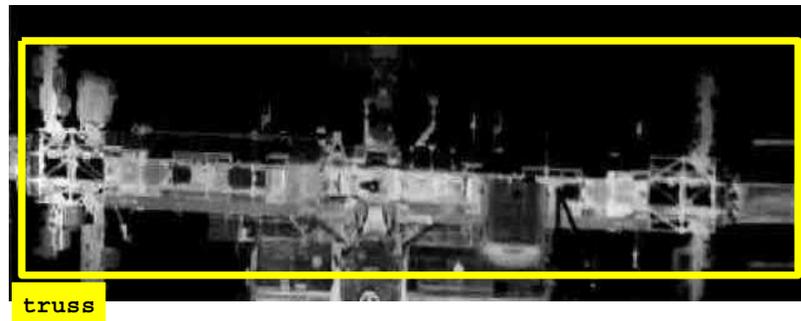
Textons	595	740	2044
FD	79	71	81
HK-0	81	71	81
HK-1	81	71	81
HK-2	17	79	81
BO	81	81	38
kNN -0	81	2	81
kNN -1	19	9	42
kNN -2	48	5	38
kNN - ∞	81	2	31
kNN - $-\infty$	62	14	33
kNN - θ	48	17	17
NB	17	36	81

Table 3.4: Mean Jaccard index comparison.

Classifiers	IoU
FD	47
HK-0	48
BO	29
kNN -2	43
NB	36



(a) module



(b) truss

Figure 3.13: Bounding boxes of the ISS query image. Subfigure (a) shows over prediction of the ISS modules. Subfigure (b) shows correct prediction of the ISS truss.

3.2 CNN Bounding Box

Deep learning ConvNet methods for object recognition and localisation can be highly precise; this work evaluates two state-of-the-art networks called *ResNet* and *Inception-ResNet* for image recognition and the *Faster-RCNN* technique for object localisation. This section provides an overview of the high-level system pipeline and details of each network.

3.2.1 Overview

Chapter 1 Sec. 1.4.2 provides a historical overview and state-of-the-art description of the ConvNet methods for recognition and localisation. In our ConvNet-based bounding box evaluation, we use the CubeSat as test example and use Python 3.5.2 with the Tensorflow 1.0.1 library as the computation platform. Our process follows the traditional deep learning networks that consist of off-line training and operation inference phases. Figure 3.14 provides an overview of these two phases. Training of the ConvNet model requires several hundred CubeSat images with Ground Truth (GT) bounding box and category ID tags. The bounding box labels are stored as *XML* files following the Pascal Visual Object Classes (VOC) [362] format^a. The GT bounding box labels and annotation files are converted into a Tensorflow binary data file or so-called *tfRecords*. The ConvNet custom settings are stored in a *Network Configuration* file; it defines the ConvNet structure and training specific parameters such as the learning rate and number of training steps. The category labels and pre-trained weights are also inputted into the network for training. The category labels provide object ID, and the pre-training weights are the network weights trained from using large image datasets such as ILSVRC [125], Pascal VOC [362], or Microsoft Common Objects in Context (COCO) [363]. The large image datasets contains hundreds of thousands to millions of annotated images, where network kernel weights are optimised to filter principal features [126]. The model weights are normally kept from the full training and *transfer learning* is applied where the final ConvNet layer is removed and retrained using the specific class labels. This fine-tuning process combines the generically trained kernel layers with the target-specific layer. Pre-trained model weights are needed for high

^a http://host.robots.ox.ac.uk/pascal/VOC/voc2012/devkit_doc.pdf

precision when the target image dataset have a low number of object categories and images. Finally, the constructed ConvNet structure, the so-called *network graph*, is frozen for inference.

During inference, we first generate realistic spacecraft movement using orbit and attitude simulation provided in Appendix B. The orbit generation process begins with initial orbit and local pose state input. Then the spacecraft orbit and attitude state are predicted by dynamic integration. We develop a forward-kinematic framework to interface with the camera emulator. Image generation is performed using the *3D Studio Max*[®] (3DS-Max[®]) software. The CubeSat Computer-Aided Design (CAD) model is imported into 3DS-Max[®] and simplified for fast rendering where non-appearing or unimportant features are removed. A virtual monocular camera is created in 3DS-Max[®] using input intrinsic camera parameters. The camera parameters were confirmed by using standard camera calibration techniques in the 3DS-Max[®] environment. When using laboratory images, the ‘*Synthetic Image Generation*’ block from Fig. 3.14 is replaced with images of the real-world camera. The camera images are evaluated using the frozen ConvNet model from training, and the resulting bounding box and probability are written into a text file.

3.2.2 ResNet

The Residual Network (ResNet) [2] was the winner of the ILSVRC-2015 recognition challenge with 3.57 percent error for top-5 image classification. Compared to the 8-layers AlexNet [86], 19-layers VGG [128], and 22-layers GoogLeNet [3], the ResNet depth has significantly increased to 152-layers. To allow for this large increase to the network depth, the ResNet changed the traditional filtered response map to store residual responses. He *et al.* [2] empirically demonstrated ConvNets degradation with increasing depth even if additional layers are constructed as identity mappings. Consider a two convolutional layer network as shown in Fig. 3.15, the output of the second layer is

$$\mathbf{y} = \mathbf{W}_2 \sigma_1 (\mathbf{W}_1 \mathbf{x} + \mathbf{b}_1) + \mathbf{b}_2, \quad (3.40)$$

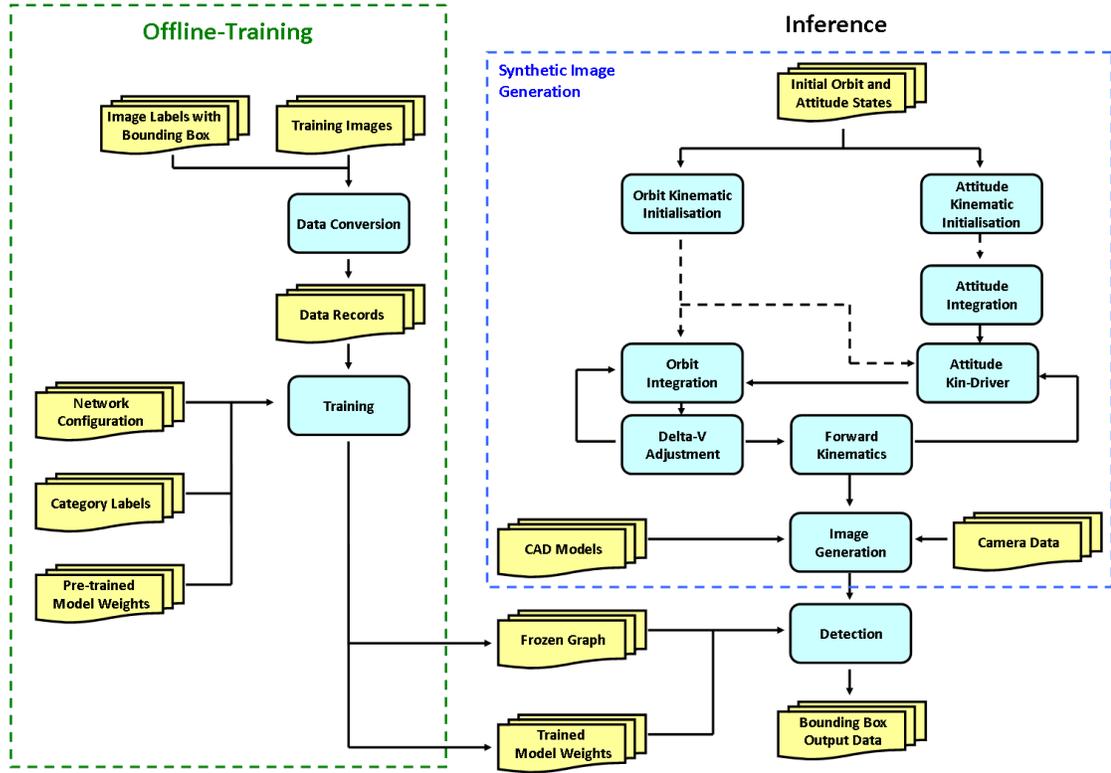


Figure 3.14: Simulation process including offline training. During inference, the dotted arrow indicates initialisation. For laboratory experiments, replace the *Synthetic Image Generation* block with real-world camera images.

where \mathbf{x} is the input layer, \mathbf{W}_i is the weights of the i^{th} layer, \mathbf{b}_i is the i^{th} layer bias, and $\sigma_i(y_i)$ is the i^{th} layer activation function. Equation 3.40 can be written as

$$\mathbf{y} = \mathcal{F}(\mathbf{x}), \quad (3.41)$$

if the input is allowed to *skip across* the two layers, then

$$\mathbf{y} = \mathcal{R}(\mathbf{x}) + \mathbf{x}, \quad (3.42)$$

such that the convolutional layers are now learning the optimal residual response in \mathbf{y}

$$\mathcal{R}(\mathbf{x}) = \mathbf{y} - \mathbf{x}, \quad (3.43)$$

Both Eq. (3.41) and Eq. (3.42) computes the optimal response map; however, it is easier to find a small fluctuation residual in \mathcal{R} than the full weights solution \mathcal{F} . Figure 3.16 provides

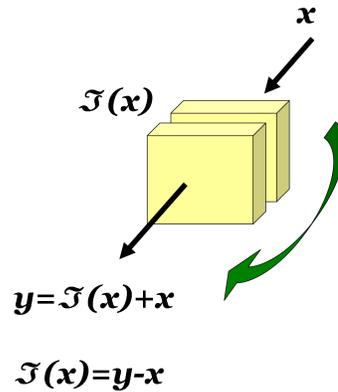


Figure 3.15: Two-layer network with *shortcut connection* to drive residual learning [2].

the full *101-layer ResNet* (ResNet-101); this network was chosen to compute the spacecraft detection problem. The final Fully Connected (FC) layer is reduced from 1000 classes to two classes for *1U_CubeSat* and *3U_CubeSat* for the orbit simulation and *red_sat* and *gauge* for the laboratory experiment.

3.2.3 Inception-ResNet

Szegedy *et al.* [4] empirically showed the combination of *ResNet* and the *Inception* modules can result in better performances and inference speed. The so-called *Inception-ResNet-V2* computes the ILSVRC top-5 image classification with 3.08 percent error [4]. The *GoogLeNet* approximates optimal local sparse structure with multiple layers of *inception* modules. The *inception* modules have parallel paths using kernels of various aperture sizes and mimics narrow and wide receptive fields. A naïve version of the inception concept is shown in Fig. 3.17(a) [3]. The final filtered response map from the filter paths concatenates as a single output tensor. The naïve model can be enhanced by adding a 1×1 filter prior to the 3×3 and 5×5 layers and after the 3×3 max pooling layer, shown in Fig. 3.17(b). The 1×1 filter collapses the input filter channels by extracting the primary features in the channel depth and greatly reduces the required computation. Finally, average pooling replaces the traditional FC layer for better accuracy. The *Inception-ResNet-V2* combines the *inception* module with the *ResNet* residual learning [4]. Larger 7×7 filters are computed

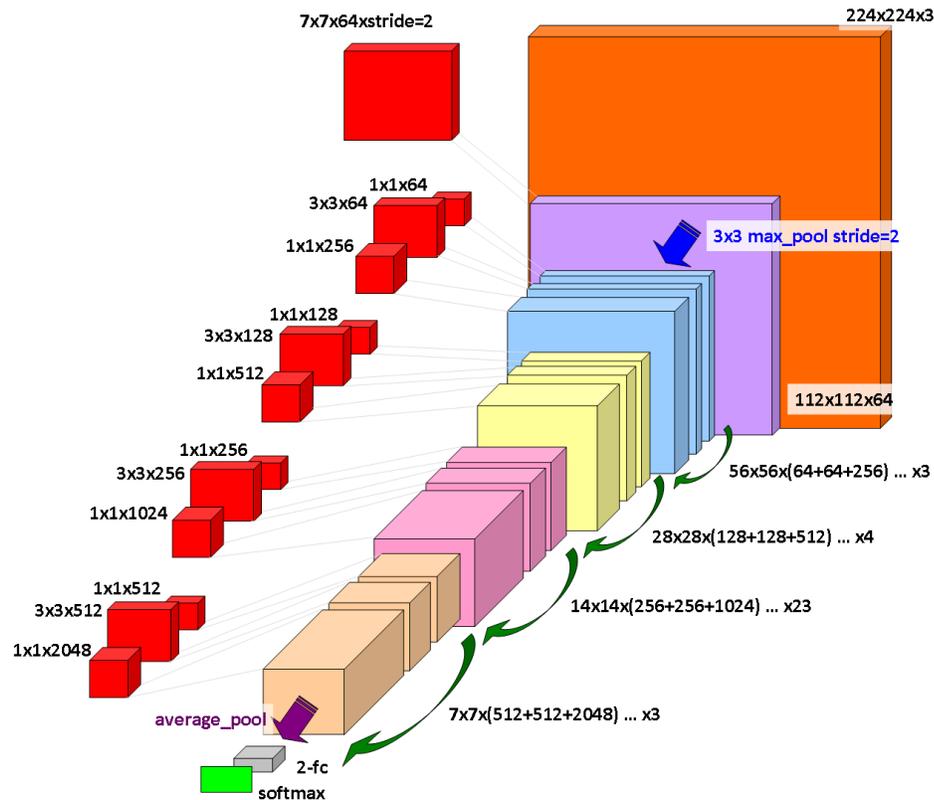


Figure 3.16: 101-Layer *ResNet* [2]. Red boxes represent filter kernels, and non-red boxes represent filter response maps and the input. The green arrows represent inputs *skip across* the residual responses. The gray box represent FC layers, and the green layer is the *softmax* classifier.

using *convolution by separability*, reducing the filter kernel into one-dimensional arrays. *Inception – ResNet* modules are introduced using the 1×1 and 3×3 inception modules with input layers that *skips across* to the output layer for addition. The *Inception-ResNet-B* module is shown in Fig. 3.18 as an example of how residual learning is performed in *Inception-ResNet*. In this example, a 7×7 kernel is separated into two one-dimensional arrays. Full details of the *Inception-ResNet* implementation can be found in Szegedy *et al.* [4]. It is one of best performing state-of-the-art ConvNet for recognition and object detection at the time of this investigation [137].

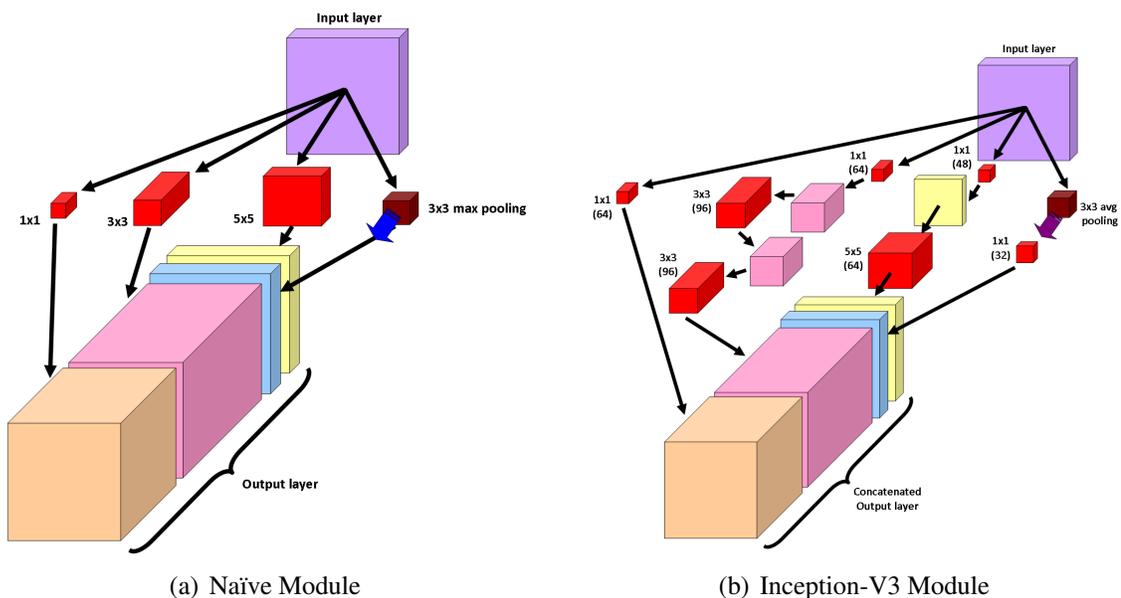


Figure 3.17: GoogLeNet Inception Module [3]. Red boxes represent filter kernels, brown boxes represent max pooling, and non-red boxes represent filter response maps.

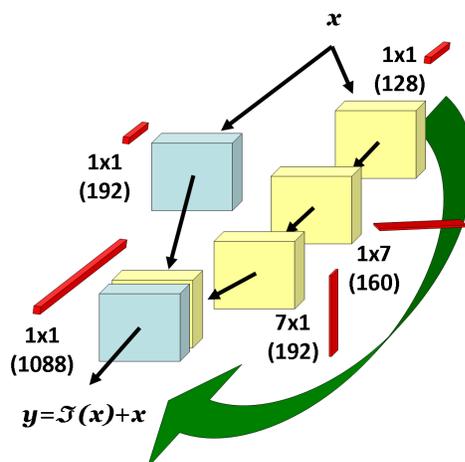


Figure 3.18: The *Inception-ResNet-V2 Inception-ResNet-B* Module [4]. Red boxes represent filter kernels, and non-red boxes represent filter response maps.

3.2.4 Object Detection

The pre-ConvNet object detection methods use SIFT and HOG features, such as those used in BoVW and DPM. While these features can be associated with cells in the visual pathway, CNN provides a more informative multi-stage system for visual classification [123]. Early ConvNet solutions to object detection are *Overfeat* [364] and the first

generation *RCNN* [131]. Although these networks are more accurate than non-CNN solutions, they generate a large number of bounding boxes by regressively merging towards the RoI. As a result of the large volume of proposals, the regression process can be highly time-consuming and requires large amounts of memory. Later improvements focused on increasing speed of localisation such as *Fast-RCNN* [132] and *Faster-RCNN* [87]. In each revision of the *RCNN* model, more efficient ways of collecting the region proposals and the use of ConvNet classification were developed.

Faster-RCNN

The *Faster-RCNN* [87] is a *Region Proposal Network* (RPN) using *ResNet-101* or *Inception-ResNet-V2* as the recognition engine; it is a FCN [146] without any FC layers. In addition to the recognition network, RPN includes a *detector head* that simultaneously regress region boundaries and objectness scores for each location in a working grid. The *detector head* was originally proposed by Girshick *et al.* [132] in *Fast-RCNN*, where the objectness score measures how well the proposal belongs to a specific class compared to the background, and the regression layer locates the object in the image. The region proposal is created by sliding a small network over the convolutional feature map of the last shared convolutional layer from the recognition network. Each sliding window is mapped to a lower-dimensional feature and fed into an FC box-regression layer plus a box-classification layer. The number of the maximum proposal at each sliding-window position is k . The regression layer has a $4k$ output for the bounding box information, and the classification layer has a $2k$ score for the probability of object or non-object in each proposal. The k proposed boxes are called *anchor* boxes. *Faster-RCNN* uses 3 scales and 3 aspect ratios, resulting in 9 *anchors* at each sliding position. Given the input feature map is $W \times H$ pixels, there are WHk *anchors* in total. The RPN contains $n \times n$ convolutional layer followed by two 1×1 convolutional layers for bounding box coordinate regression and object classification scores.

The loss function for *Faster-RCNN* is

$$L = \frac{1}{N_c} \sum_i L_{ci} + \lambda \frac{1}{N_r} \sum_i p_i^* L_{ri}, \quad (3.44)$$

where c and r represent classification and regression respectively; i is the index of an anchor in the mini-batch, N_c is the mini-batch size, N_r is the number of anchor locations (*i.e.* WH). The anchor classification loss, L_{ci} , is computed using the ground-truth label and the anchor's predicted probability. The anchor label is positive if the highest IoU overlaps with the ground-truth bounding box, or if the anchor IoU with respect to the ground-truth bounding box is greater than 0.7. The anchor is negative if the IoU with all ground-truth boxes is less than 0.3. The ground-truth anchor label (p_i^*) is 1 if the anchor is positive, and 0 otherwise. Anchors that are neither positive or negative are omitted from training; λ is set to let N_r and N_c to have a similar order of magnitude. The regression loss, $p_i^* L_{ri}$, is activated if $p_i^* = 1$ and $L_{ri} = R(t_i - t_i^*)$ where R is the L_1 loss function for improving robustness [132],

$$R = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise,} \end{cases} \quad (3.45)$$

The parameters for bounding box regression are,

$$\begin{aligned} t_x &= (x - x_a) / w_a, & t_y &= (y - y_a) / h_a \\ t_w &= \log(w / w_a), & t_h &= \log(h / h_a) \\ t_x^* &= (x^* - x_a) / w_a, & t_y^* &= (y^* - y_a) / h_a \\ t_w^* &= \log(w^* / w_a), & t_h^* &= \log(h^* / h_a), \end{aligned} \quad (3.46)$$

where x , y , w and h is the bounding box centre coordinates and the width and height respectively; x , x_a , and x^* represent the predicted box, anchor box, and ground truth respectively, similar is true for y , w , and h . The *Faster-RCNN* RPN is shown in Fig. 3.19.

3.2.5 Metric Description

Synthetic camera images are generated using the 3DS-Max[®] animation software. The simulated spacecraft pose is entered into 3DS-Max[®] for motion rendering. Figure 3.20 depicts the 3DS-Max[®] work environment, the Client Satellite (CS), and the Servicer Spacecraft (SS). For simplicity, the SS camera is positioned in the SS body frame and has the same orientation. A virtual camera was created in 3DS-Max[®], the camera intrinsic-matrix was calibrated using standard techniques in the virtual environment. The camera intrinsic-matrix

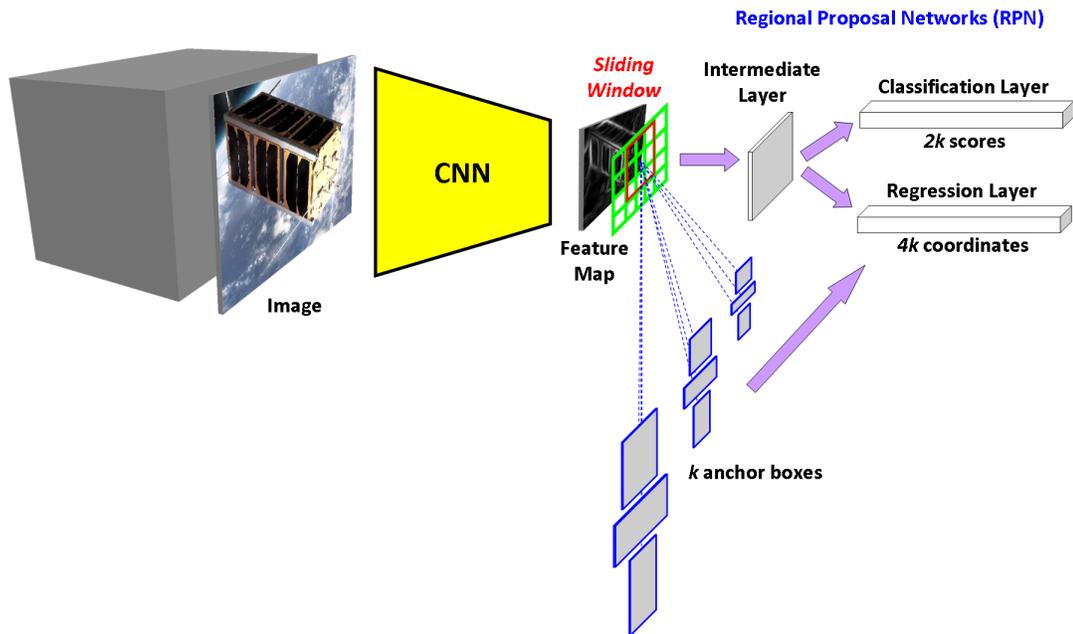


Figure 3.19: Faster-RCNN Region Proposal Network.

\mathbf{K} has the following values:

$$\mathbf{K} = \begin{bmatrix} 3,544.416 & -1.874 & 320 \\ 0 & 3,532.546 & 240 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.47)$$

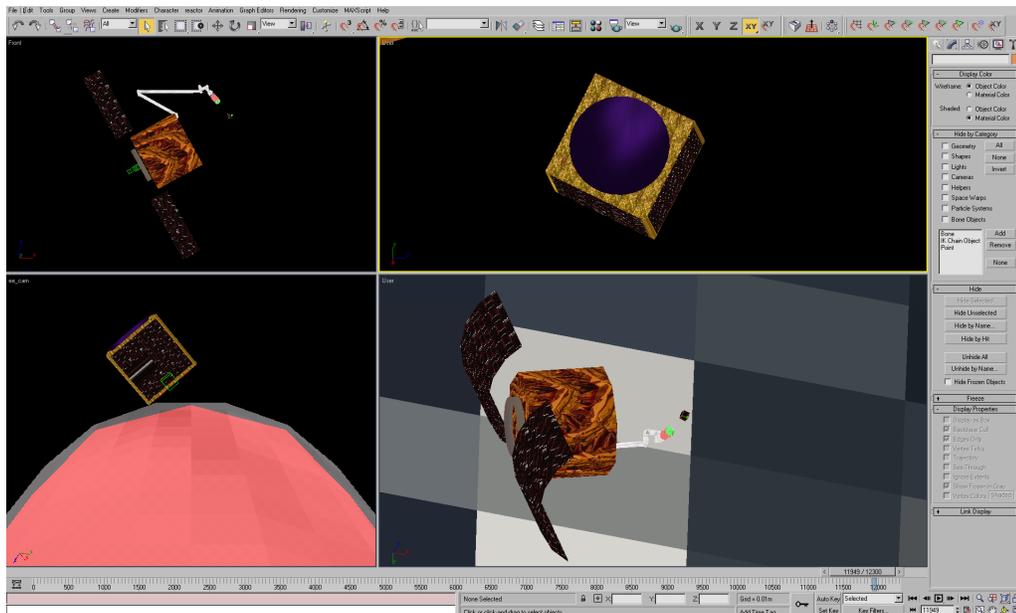
A mixture of real and synthetically generated CubeSat images was used for training and inference, the description of the CubeSat and CubeSat image dataset used are provided in Table 3.5.

Experimental images are captured from the Spacecraft Proximity Operations Testbed facility of Carleton University Spacecraft Robotics and Control Laboratory (SRCL) shown in Fig. 3.21; this facility provides dynamic spacecraft hardware simulation using a gravity offset table and air-puck floatation activated spacecraft platforms. The platform hardware parameters and ConvNet class list, training and inference image dataset size are provided in Table 3.6. The experiment images were captured using an *iPhone-6-A1549*[®] *iSight*[®] camera.

The IoU, or so-called *Jaccard index*, from Sec. 3.1.4 Eq.(3.39) measures the bounding

Table 3.5: Virtual camera parameters, HFOV is Horizontal Field of View.

Description	Data
1U CubeSat Geometry (cm)	10×10×10
3U CubeSat Geometry (cm)	10×10×30
Class List	1U_CubeSat, 3U_CubeSat
Virtual Camera Focal Length (mm)	200
Virtual Camera HFOV (deg)	10.286
Image Dimensions	640×480
Total Number of Images	500
Number of Real CubeSat Images	39 (8%)
Number of Training Images	400 (80%)
Number of Inference Images	100 (20%)

**Figure 3.20:** 3D Studio Max[®] environment for synthetic image generation.

box prediction performance. An IoU threshold greater than 0.5 indicates an Actual Positive (AP) bounding box match [362]. When the probability score for the bounding box is above 0.5, the bounding box is considered an Estimated Positive (EP). The True Positive (TP) is defined as $TP = AP \cap EP$. We use the ROC [351] accuracy, recall and precision metrics $a = (TP + TN)/N$, $r = TP/AP$, $p = TP/EP$ respectively; where TN is the True Negative and N is the number of images. The standard evaluation for ConvNet performance is

Table 3.6: SRCL spacecraft platform parameters.

Description	Data
Testbed Geometry (cm)	30×30×30
Testbed Surface Material	acrylic
Class List	red_sat, gauge
<i>iSight</i> [®] Camera Specification	8 MPix, f/2.2
Image Reduction	640×480
Total Number of Images	556
Number of Training Images	500 (90%)
Number of Inference Images	56 (10%)

**Figure 3.21:** Carleton University Spacecraft Robotics and Control Laboratory.

by using the mean-Average-Precision (*mAP*) [362] defined as

$$\begin{aligned}
 mAP &= \frac{1}{C} \sum_{i=1}^C AP_i \\
 &= \frac{1}{C} \sum_{i=1}^C \left[\int_0^1 p_i(r) dr \right] \\
 &\approx \frac{1}{C} \sum_{i=1}^C \left[\frac{1}{11} \sum_{r \in \{0,0.1,\dots,1\}} \tilde{p}_i(r) \right],
 \end{aligned} \tag{3.48}$$

where C is the number of classes and \tilde{p}_i is the *interpolated* precision for the i^{th} class. The precision interpolation is performed by taking the maximum precision measured over the

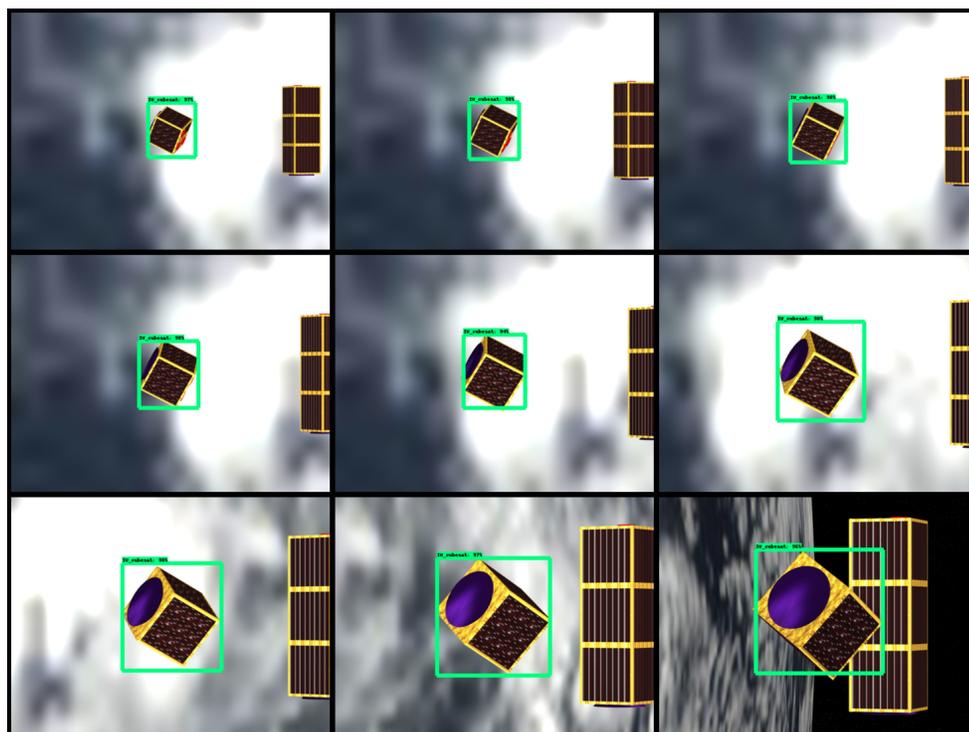
corresponding recall exceeding r as follows,

$$\tilde{p}(r) = \max_{\tilde{r}:\tilde{r}\geq r} p(\tilde{r}). \quad (3.49)$$

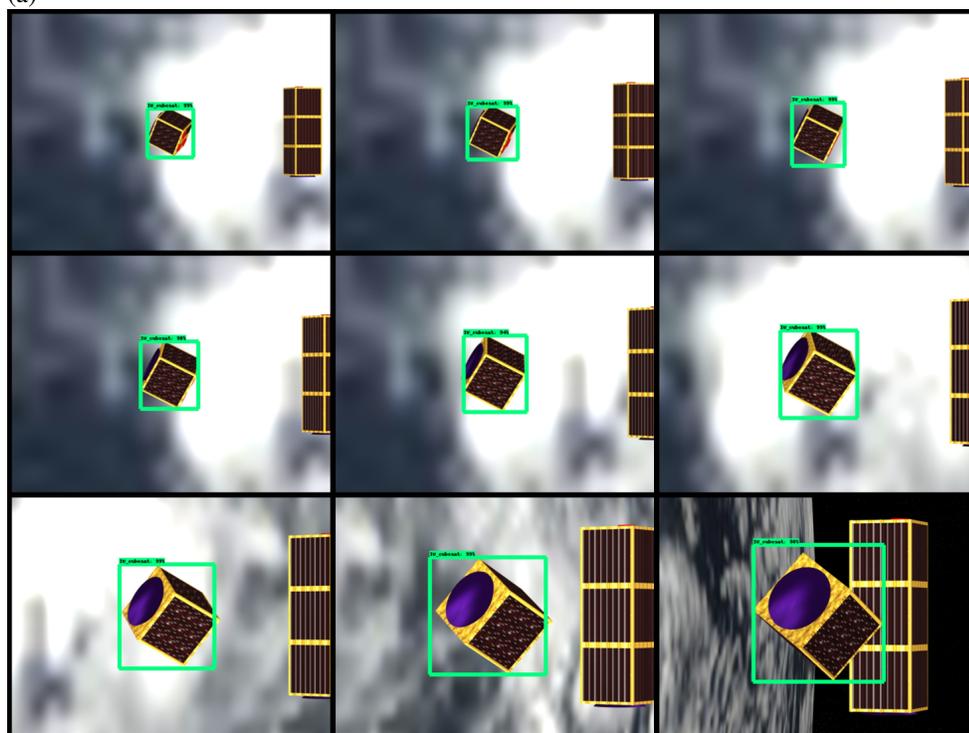
3.2.6 Localisation Performance

Qualitative results of the CubeSat detection is provided in Fig. 3.22, both *Inception-Resnet-V2* and *ResNet-101* produced excellent detection and bounding box generation. In addition to the synthetic results, two sets of laboratory experiment are provided in Fig. 3.23 and Fig. 3.24 under light and dark lighting conditions respectively. The *CubeSat* class detection has high probability while the *Guage* and *3U-CubeSat* were not detected in most images. The reason is due to a lack of training images for these secondary class objects. Both networks can become unstable during training and can saturate quickly. Part of the cause for the instability is possibly due to a lack of different training image views resulting in over-fitting. A large regularisation weight was applied to stabilise the training. The results from longer training sessions were not as accurate as short training sessions soon after the initial loss decay. The model weights trained using short duration worked sufficiently well.

Only the main CubeSat body is the subject of interest for navigation state estimation; the number two class objects are omitted in the evaluation. Quantitative results for ROC, average inference timing, mAP and average IoU for class 1 objects are provided in Table 3.7 and Table 3.8 respectively. The ROC results in Table 3.7 is based on detection probability and IoU threshold of 0.5. In summary, *Inception-ResNet-V2* out-performs *ResNet-101* in accuracy and precision in the Synthetic Images. On the other hand, *ResNet-101* resulted in higher accuracies in the lab experiment images. Overall, *Inception-ResNet-V2* has higher accuracy and precision than *ResNet-101*. Additionally, lab experiment images have higher accuracy and precision than the synthetic ones. The detection probability score and IoU thresholds were varied between zero to one independently for a comprehensive comparison between *Inception-ResNet-V2* and *Resnet-101*. The mAP from the precision versus recall curve and the average IoU are evaluated over all images. Table 3.8 shows *Inception-ResNet-V2* outperforms *ResNet-101* in both mAP and \overline{IoU} by less than one percent; however, the *Inception-ResNet-V2* inference time is nearly four times *ResNet-101* and the training time

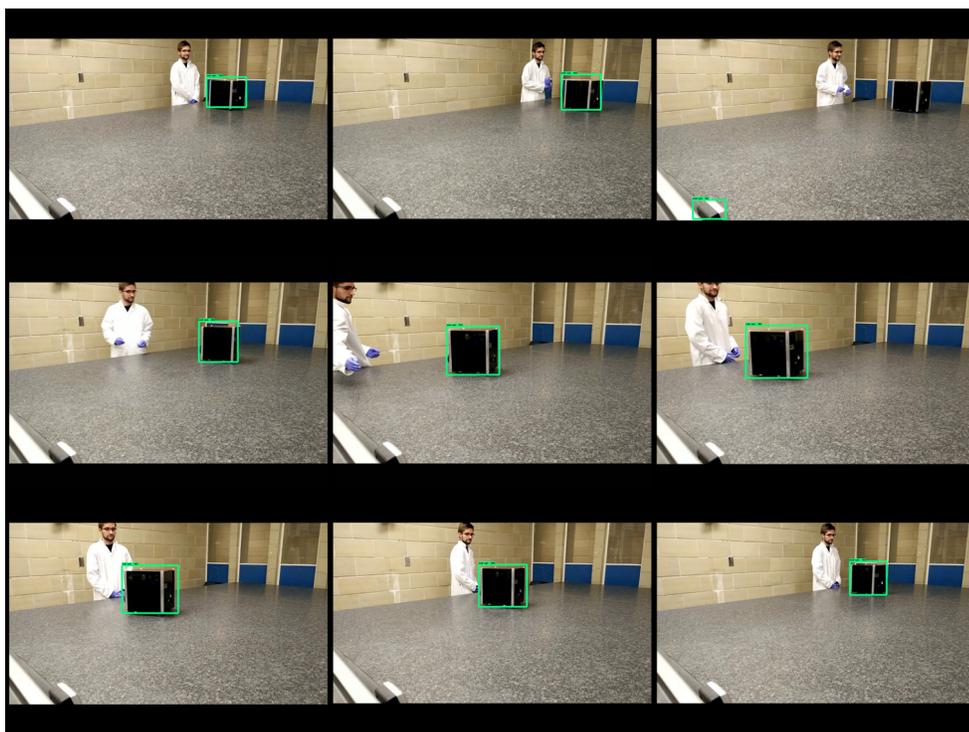


(a)

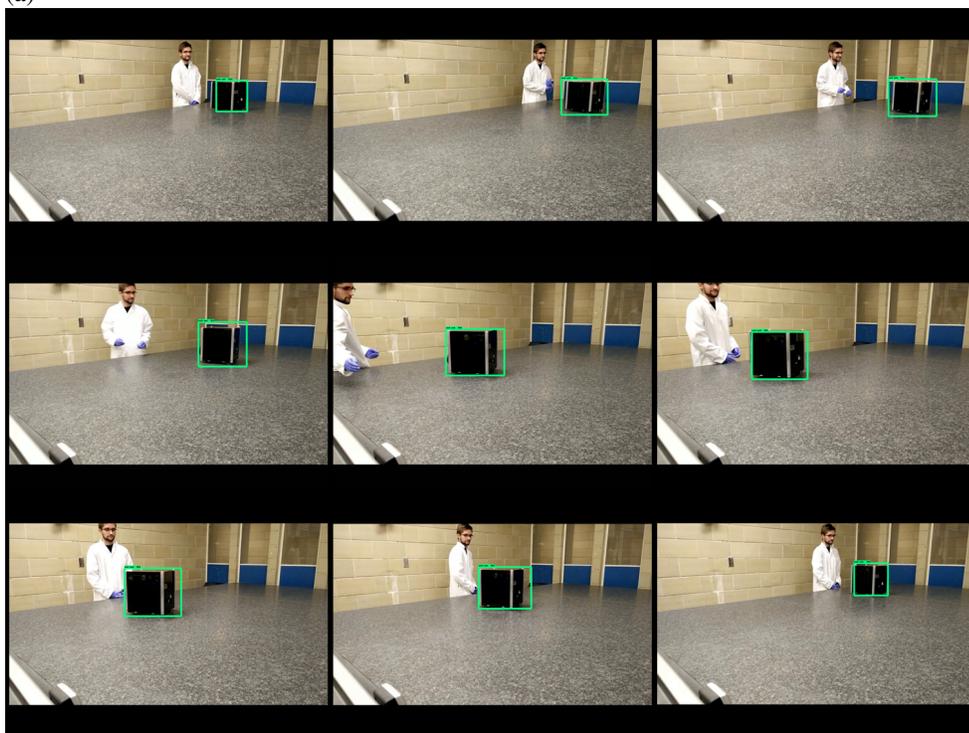


(b)

Figure 3.22: Simulated CubeSat detection using faster-RCNN based on various classifier networks. Subfigure (a) is using the *Inception-ResNet-V2* network and Subfigure (b) is using the *ResNet-101* network. <https://youtu.be/AfBw4jGBz6Y>

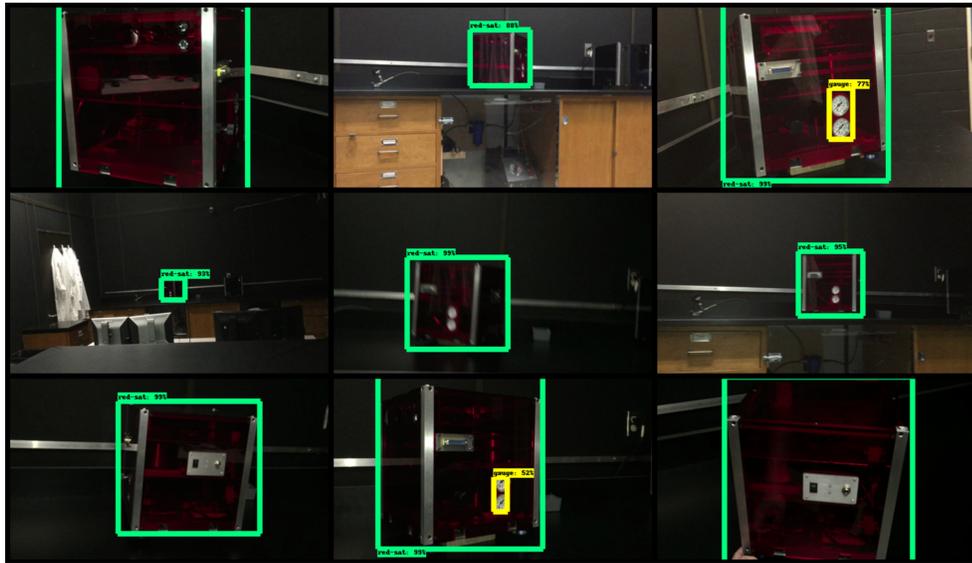


(a)

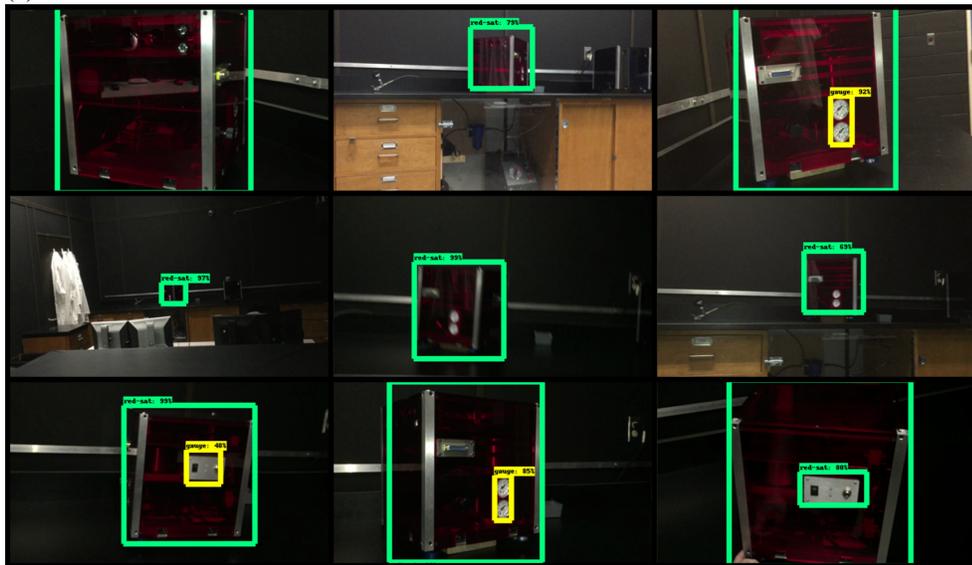


(b)

Figure 3.23: CubeSat test platform detection using faster-RCNN using different classifier networks. Subfigure (a) is using the *Inception-ResNet-V2* network and Subfigure (b) is using the *ResNet-101* network. <https://youtu.be/M1ceJWM4pKM>



(a)



(b)

Figure 3.24: Laboratory environment CubeSat platform detection using faster-RCNN based on various classifier networks. Subfigure (a) is baselined in *Inception-ResNet-V2* and Subfigure (b) is baselined in *ResNet-101*. <https://youtu.be/GwaIWk6cjzs>

more than doubles *ResNet-101*. Figure 3.25 provides a detailed analysis of network performance. Figure 3.25(a) shows IoU versus $1 - \text{Detection Probability}$. By displaying the probability error, the detection probability can be plotted on a semi-log graph exposing more differences between the *Inception-ResNet-V2* and *ResNet-101* networks. Majority of the bounding box detection have a probability higher than 0.9 with IoU higher than 0.7. The *Inception-ResNet-V2* detection probability error is lower than 10^{-4} to 10^{-3} in the lab experiment images. Figure 3.25(b) shows both networks have very high precision versus recall performance, where *Inception-ResNet-V2* enjoys a slight advantage over *ResNet-101* when the recall rate is nearly one. Figures 3.25(c) and (d) shows *Inception-ResNet-V2* outperforms *ResNet-101* in accuracy for changing detection probability and IoU threshold; however, when the detection probability threshold is above 0.9, the accuracy between the two networks is nearly identical. In summary, the *Inception-ResNet-V2* is a more accurate network than *ResNet-101* due to advantages of narrow and wide receptive fields. For the same reason, the *Inception-ResNet-V2* is slower to operate and takes longer to train.

Table 3.7: Network performance for Class 1 objects. All results are computed using the inference datasets for synthetic and lab images. Comparing synthetic and real images and various network types. Detection Probability and IoU thresholds set to 0.5

Description	Accuracy	Recall	Precision
Synthetic Images <i>Inception-ResNet-V2</i>	0.9495	1.0000	0.9265
Synthetic Images <i>ResNet-101</i>	0.8081	1.0000	0.7683
Real Images <i>Inception-ResNet-V2</i>	0.9636	0.9630	1.0000
Real Images <i>ResNet-101</i>	1.0000	1.0000	1.0000
<i>Inception ResNet-V2</i>	0.9545	0.9829	0.9583
<i>ResNet-101</i>	0.8766	1.0000	0.8593
Synthetic Images	0.8788	1.0000	0.8400
Real Images	0.9818	0.9813	1.0000

The *ResNet* and *Inception-ResNet* with *faster-RCNN* were used to compute real-time bounding boxes around a target spacecraft. ConvNet spacecraft image transfer learning is needed to fine-tune the generic network to recognise the target vehicle. The experiments were conducted using synthetic and real-world laboratory images. Results show the *faster-RCNN* network based on *Inception-ResNet-V2* and *ResNet-101* image classifiers produce highly precise localisation of the target spacecraft vehicle. The *Inception-ResNet-V2* network is slightly more precise, while the *ResNet-101* network can be trained in less than

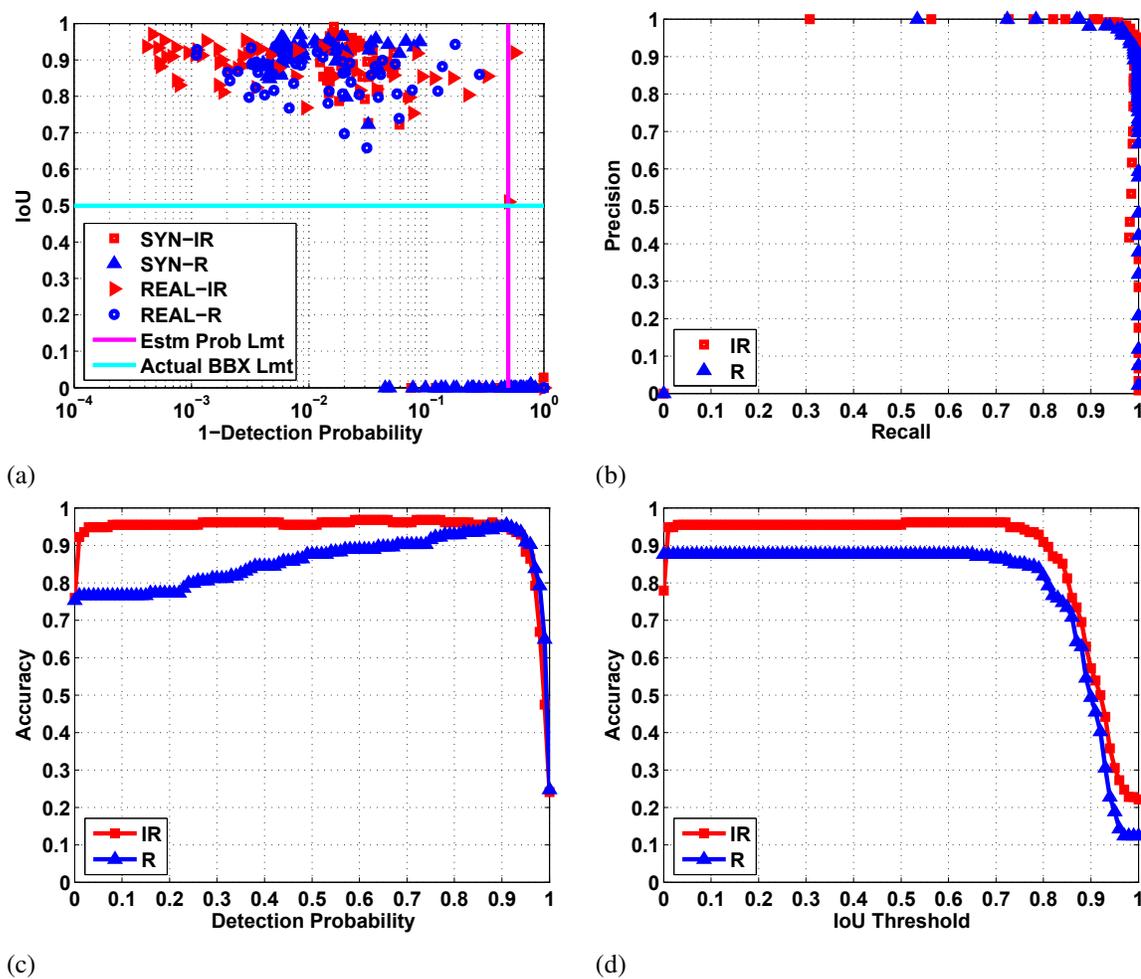


Figure 3.25: Network performance for Class 1 objects. All results are computed using the inference datasets for synthetic and lab images; SYN is the synthetic CAD images from the CubeSat ProxOps simulation; REAL is the experimental lab images of the spacecraft platforms; IR is the *Inception-ResNet-V2* network model; R is the *ResNet-101* network model; *Estm Prob Lmt* is the probability threshold of 0.5, detection box with probability lower than this threshold is rejected; *Actual BBX Lmt* is the IoU threshold of 0.5, bounding boxes with IoU higher than this threshold are considered as Actual Positives (*AP*). Subfigure (a) provides IoU for various test conditions and network configurations. Subfigure (b) compares precision versus recall for *Inception-ResNet-V2* and *ResNet-101* models by varying both detection probability and IoU thresholds. Subfigure (c) and (d) compares network accuracy of varying detection probability and IoU thresholds. When computing accuracy, the non-varying threshold is held to the default value of 0.5 for both IoU and detection probability. Graphs are best viewed in colour.

Table 3.8: Network performance for Class 1 objects. Comparing various network types with threshold variation. \overline{IoU} denotes average IoU over test images omitting images with $IoU = 0$.

Description	Average Training Timing (min)	Average Inference Timing (ms)	mAP	\overline{IoU}
<i>Inception-ResNet-V2</i>	30.0	418	0.9956	0.8814
<i>ResNet-101</i>	12.1	115	0.9880	0.8805

half the time and is nearly four-times faster during inference. The biggest drawback of the CNN network is the training stability and requiring hundreds of labeled images. Too many training images that resemble each other will cause training instability. Real-time object localisation can be achieved with both tested networks.

Chapter 4

Background Removal

Bounding boxes reduces the image to a smaller RoI. The smaller rectangular RoI, however, still contains background images and do not precisely outline the target border. Pixel-wise classification or foreground extraction will produce precise target border outline, three popular methods to achieve this are background subtraction, semantic segmentation, and image saliency detection. In the following sections, we develop new foreground extraction methods, specifically, compact autoencoder ConvNets and novel saliency generation. The developed techniques are compared to traditional and state-of-the-art methods.

4.1 CNN Segmentation

Convolutional Neural Network, or ConvNet, is a type of deep neural network with optimised image filter layers to learn elementary image structures on various perception levels. The network reduces in image size and increases in channel depth to perform recognition; it can also reverse its size and depth to learning pixel-wise classification. This section provides the description and training procedures of the ConvNet models for semantic segmentation and will discuss the evaluation results.

4.1.1 Network Description

Our ConvNet semantic segmentation work is performed in two phases; first, five ConvNet recognition models are compared for their performances. In the first phase, we trained and evaluated the CalTech-101 and our Space-5 image datasets to gain experience with each network behaviour, to tune network hyperparameters, and to build the network weights from scratch for recognition. In the second phase, we selected three out of the five networks to generate the segmentation images. A mirror decoding network is added to the encoding half removing the FC dense layers, and the resulting network becomes an FCN. A set of ISS images with manually labeled segmentation masks are used for training in the second

phase. Three finally selected autoencoder networks produced the semantic segmentation map during inference and the resulting images are saved to file. The full training and evaluation pipeline are shown in Fig. 4.1, all networks are built using *TensorFlow* 1.0.1 and *Python* 3.5.2.

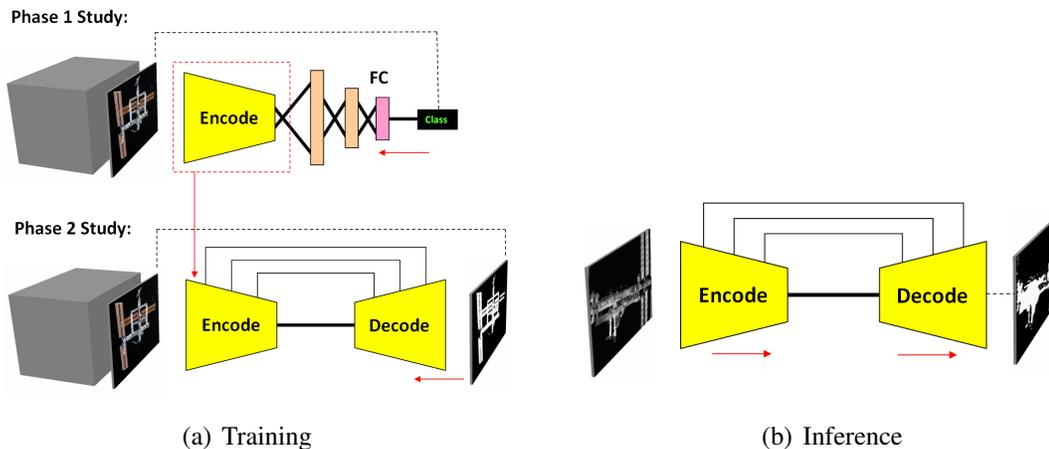


Figure 4.1: Training and evaluation pipeline, refer to Table 4.1 for encoding network details. Decoding network mirrors the encoding network. Phase 1 network is trained for single class identification. Phase 2 transfers the network weights and batch normalisation parameters from phase 1. The encoder network parameters and weights are frozen during phase 2 training. A batch of 128 images forms the input tensor and is fed through the FCN with labelled target masks. Feature map tensors from the first convolutional layer of every encoder network groups are *skipped* to the respective decoder side. During inference, the FCN runs forward to generate the semantic segmentation map.

We evaluate five network models for their classification performance on CalTech-101 and Space-5 image datasets. Namely, the networks are: *AlexNet-5*, *AlexNet-8*, *UNet-8*, *VGG-19*, and *Darknet-21*. The number after each network indicate the number of encoder convolutional and fully connected layers. All networks use maxpooling to downsample the feature map after each convolution groups. Batch normalisation [365] is applied after each convolutional layer followed by leaky Rectified Linear Units (ReLU) [366] activation using a slope of 0.2. Backpropagation is performed using *stochastic gradient decent* (SGD) with an initial learning rate of 0.1, VGG and Darknet initial learning rate is 0.05, and learning rate decay of 0.1 after every 20×10^3 iterations. The dropout rate is 0.9, training and validation batch size is 128, and convolution stride is 1 unless otherwise specified. The detailed architecture of each network is in Table 4.1.

Our implementation of the AlexNet [86] retains most network features with some minor improvements recommended by recent ConvNet developments. The AlexNet [86] uses ReLU [367] as activation instead of the traditional sigmoid function. It also uses the dropout [368] technique to minimise overfitting. Krizhevsky [86] splits the convolutional channels so they can run independently on two GPUs and added Local Response Normalisation (LRN) to each convolutional layer for brightness normalisation. *AlexNet-5* is a simplified version of the baseline AlexNet, it contains only two convolutional layers and three fully connected dense layers with max pooling and LRN after each convolutional layer. This lightweight network requires lower memory and is ideal for small datasets like CIFAR-10 [369]. *AlexNet-8* is a modified AlexNet with batch normalisation and leaky ReLU activation in all convolutional layers. GPU specific convolutional layers and LRN were omitted from our implementation since they do not add increase performance [128]. The five convolutional layers step down the input image from 120×120 to 30×30 to 15×15 to 8×8 using max pooling at the end of each convolutional layer group. *AlexNet-8* uses 4 pixel stride in the first convolutional layer resulting in the largest reduction of feature map dimension.

Ronneberger *et al.* [147] developed UNet with a five group auto-encoder FCN each with two convolutional layers and maxpooling. We modify the UNet to three groups with one layer in the first and last group to reduce memory. The UNet uses *skip layer* connections to maintain feature map structure [154]. Our *UNet-8* network consists of five convolutional layers and three fully connected layers and *skip layer* connections. *VGG-19* [128] is a popular network that is sized between AlexNet [86] and GoogLeNet [3]. The baseline *VGG-19* [128] consists of two groups of two convolutional layers with 64 and 128 channels, and three groups of four convolutional layers with 256, 512 and 512 channels. We use a modified *VGG-19* where the ConvNet channels are 16 – 32 – 64 – 128 – 256 for each ConvNet group respectively to reduced memory.

The YOLO *Darknet* [134] was inspired by the GoogLeNet [3] but does not have inception modules. It uses three to five alternating 1×1 and 3×3 kernels based on the *network-in-network* [370] structure. The YOLO *Darknet* [134] consists of seven groups with 24 convolutional layers, it contains one convolutional layer in the first two groups and max-pooling in the first four groups. The YOLO-9000 *Darknet* [135] is simplified to six groups

with first five groups ending in max pooling. The YOLO network also contains one fully connected layer and one detection tensor to predict bounding box localisation. Our *Darknet* omits the detection layer and uses the same fully connected dense layer as the *AlexNet*. All the tested networks are connected to three FC dense layer at the end of the convolutional groups for classification. *AlexNet-5*, *UNet-8*, and *VGG-19* using roughly 400 and 200 nodes for the first two layers while *AlexNet-8* and *Darknet-21* using roughly 200 and 100 nodes. Due to small datasets used in training, the FC nodes were reduced by a factor of 10 – 20 from the baseline networks. The number of nodes in the final *SoftMax* layer equals the total number of classes in the dataset.

Long *et al.* [146] demonstrated the semantic segmentation map could be generated using FCN auto-encoders with encoding and decoding networks. We followed the DeconvNet and Segnet approach of using mirrored auto-encoders. In the decoding network, reverse max pooling is typically computed by restoring the forward max pooling coordinates in the upsampling layer [126]. A more efficient approach is to let the stride convolution learn its own spatial upsample [371]. Our approach also adapts the *skip layer* concept from UNet to allow structured response map generation [154]. *Skip layers* are feature maps that ‘skipped’ from the encoder side to concatenate with decoding maps of roughly the same size. In the case where the *skip layer* does not match the decoder response map size, the *skip layer* feature map is randomly cropped. The cropping shall not typically exceed ten pixels and is only one to two pixels in the deep channel layers. During training, the pixel-wise class errors from the ground truth labels are summed as one loss value to be minimised during backpropagation.

4.1.2 Image Datasets

We use CalTech-101 [372] and the Space-5 dataset for training and inference. The CalTech-101 dataset consists of 102 category objects provided by 9,144 images of various sizes with typical resolution of 300×200 . This work randomly selects ten percent of the images for inference and use the rest for training. The CalTech-101 dataset is compact and easy to use but do not have pixel-wise data labels. There are numerous image datasets available for ConvNet studies; however, there is no dataset specifically developed for spacecraft vision navigation. To this end, we developed a new spacecraft image dataset called *Space-5*

Table 4.1: Network architectures; the first three numbers represent height, width, and channel of each feature layer, -sign indicates Batch Normalisation (BN) and Leaky ReLU (LR) combination layers. +sign indicate BN, LR and Drop Out (DO) combination layers. The value after ‘c’ indicates square kernel size, the value after ‘s’ indicates the stride, the value after ‘w’ indicates the window size, FC indicates Fully Connected, LRN indicates Local Response Normalisation.

AlexNet-5	AlexNet-8	UNet-8	VGG-19	Darknet-21
	input(120×120×3 RGB image)			
120×120×64-c5-	30×30×48-c11-s4-	120×120×16-c11-	120×120×16-c3- 120×120×16-c3-	120×120×16-c7-
maxpool-w3s2 /LRN	maxpool-w2s2			
60×60×64-c5-	15×15×128-c5-	60×60×32-c5- 60×60×32-c5-	60×60×32-c3- 60×60×32-c3-	60×60×32-c3-
LRN /maxpool-w3s2	maxpool-w2s2			
	8×8×192-c3- 8×8×192-c3- 8×8×128-c3-	30×30×64-c3- 30×30×64-c3-	30×30×64-c3- 30×30×64-c3- 30×30×64-c3- 30×30×64-c3-	30×30×64-c3- 30×30×32-c1- 30×30×64-c3-
	maxpool-w2s2			
			15×15×128-c3- 15×15×128-c3- 15×15×128-c3- 15×15×128-c3-	15×15×128-c3- 15×15×64-c1- 15×15×128-c3-
			maxpool-w2s2	
			8×8×256-c3- 8×8×256-c3- 8×8×256-c3- 8×8×256-c3-	8×8×256-c3- 8×8×128-c1- 8×8×256-c3- 8×8×128-c1- 8×8×256-c3-
			maxpool-w2s2	
				4×4×512-c3- 4×4×256-c1- 4×4×512-c3- 4×4×256-c1- 4×4×512-c3-
FC-384+	FC-205+	FC-384+	FC-410+	FC-205+
FC-192+	FC-102+	FC-192+	FC-204+	FC-102+
FC-(Number of Classes)/SoftMax				

using a mixture of 4,237 synthetic and real images of five object categories: Earth (230 images), ISS (2,590 images), Spacecraft (193 images), Envisat (301 images), and the RSM (421 images). Sample images of each class category are provided in Fig. 4.2. Segmentation annotation for the ISS and the Earth were manually generated, the Neptec *TriDAR* IR video of the SSO STS-135 mission undocking sequence was used. During the STS-135 undocking sequence, the SSO performs a flyby of the ISS with the rotating Earth in the background. For segmentation training and inference, the Space-5 dataset is reduced to the ISS and Earth (Space-2). More inference images were added to Space-2, including 10 Earth only images and 141 ISS only IR images from the STS-135 mission.

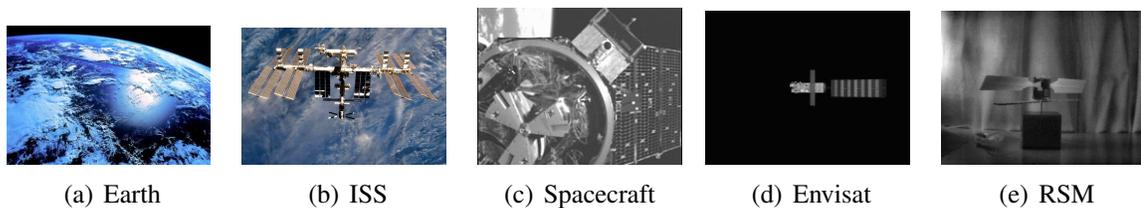


Figure 4.2: Sample images from the Space-5 database.

4.1.3 Evaluation Metrics

We use the same IoU and ROC definitions as in Chapter 3, Secs. 3.1.4 and 3.2.5. The pixel accuracy defined by Long *et al.* [146] is equivalent to the ROC precision. In addition to the aforementioned metrics, the total and mean pixel accuracy, total and *mean IoU*, and *frequency weighted IoU* (J_{fw}) from Long *et al.* [146] are also computed. The total pixel accuracy is defined as $Prec = \sum_i TP_i / \sum_i EP_i$, the mean pixel accuracy is defined as $\overline{Prec} = \sum_i Prec_i / n_{CL}$, where n_{CL} is the total number of class objects. The mean IoU is $\overline{J} = \sum_i J_i / n_{CL}$ and J_{fw} is defined as $J_{fw} = \sum_i \frac{EP_i TP_i}{EP_i + AP_i - TP_i} / \sum_i EP_i$. Finally, the mean accuracy (\overline{Acr}) and recall (\overline{Rcal}) are averages over the number of classes.

4.1.4 CNN Segmentation Performance

Results of the network classification for the CalTech-101 and Space-5 dataset are provided in Table 4.2. The Space-2 dataset validation resulted in 100 percent accuracy for all five networks. The CalTech-101 and Space-5/2 dataset training were all conducted with 60×10^3 batch iterations. The computed accuracy is lower for the higher class datasets due to lack

of training images. Zeiler [126] proposes to train deep ConvNets on high volume datasets such as ImageNet [125] with one million images and one thousand classes to avoid overfitting and optimise filter layers. Applying DO to only FC layers resulted in roughly 4 percent increase in accuracy from applying DO on all network layers. In general, Darknet outperforms the other networks but is the most expensive to compute.

Table 4.2: Classification results, +sign indicate dropout added to convolutional layers.

Database	AlexNet-5	AlexNet-8	UNet-8	VGG-19	Darknet-21
CalTech-101+	56.6	58.9	48.8	57.0	56.1
CalTech-101	57.9	60.0	54.9	59.3	60.4
Space-5+	96.5	96.1	94.9	94.9	95.7
Space-5	97.3	97.1	97.3	96.5	97.7

The *Darknet-21* is memory and computationally intensive despite its higher recognition performances. *AlexNet-5* was the least memory and computation intensive, but has lower performance than *AlexNet-8* on the CalTech-101 dataset; therefore, *AlexNet-8*, *UNet-8* and *VGG-19* were selected for the phase 2 semantic segmentation. The encoder weights from phase 1 are trained to 60×10^3 iterations (*i.e.* 2,727 epochs). All attempts to simultaneously train the encoder and decoder networks from scratch resulted in instability. Phase 2 convergence was much slower than Phase 1 largely due to a small learning rate of 10^{-8} is needed for stability. The *UNet-8* learning rate was reduced to 0.5×10^{-8} at the mid-point of training to avoid instability. The learning rate magnitude must sustain a stable gradient, which is the sum of all the pixel losses. Learning decay was omitted because the loss profile remains shallow but constant. All networks were trained for 600×10^3 iterations (*i.e.* 27,273 epochs) on separate GPUs. The hardware specification are as follows: CPU: AMD X8 FX-8350 8-Core Sock at AM3+ 4GHz, GPU-1: GeForce GTX-1080 11 GB, GPU-2: GeForce GTX-1080 8 GB. Phase 1 training was on the order of four to six hours for each network, and Phase 2 training took roughly five days.

Figure 4.3 provides the segmentation results of two sample ISS images. Table 4.3 provides the evaluation metrics. All networks located the primary target body and *UNet-8* produces the most accurate result. Earth prediction results are much lower than ISS due to a smaller and coarse training set. One may increase the image volume to improve network accuracy by training with the ImageNet dataset [125]. Other improvements may result

from refining the hyperparameters and increasing the input resolution. While it is visually evident *UNet-8* outperforms the other networks in Fig. 4.3, it is not immediately intuitive for UNet’s higher performance in the photo image. A superficial observation of Fig. 4.3 suggest AlexNet better predicts the ISS. This discrepancy can be explained by considering all of the classes in the scene since the presented metrics accounts the average. As a result, while the true positive of ISS in the UNet image is low, the union of space pixels are also low and cause the space IoU to be higher and the average metric in UNet to outperform AlexNet and VGG. The ISS only pixel precision of 0.63, 0.52, and 0.57 respective to AlexNet, UNet, and VGG confirms UNet as the worst ISS predictor. The above discussion highlights the importance in the correct interpretation of the metric results. In actual operations, the ISS prediction is far more important than Earth and space; therefore, it is more suitable to use weighted accuracy and IoU metric based on operational needs. Timing wise, *AlexNet-8* is the fastest, and the deeper *VGG-19* is the most computationally expensive. GPU inference can be 12, 37 and 55 times faster than CPU for AlexNet, UNet, and VGG respectively. Qualitative results of ISS extraction is provided in Fig. 4.4. The figure shows the original video sequence, a manual segmentation based on edge detection and dilation, all *OpenCV 3* background subtraction methods and the adaptive method are described in Chapter 1 Sec. 1.4.2. Contrary to ConvNets, background subtraction is mostly dependent on the relative motion and can be restrictive. The best ConvNet results are frame 1, 164, and 219; the worst is frame 55. In frame 55, the pixel intensities of the ISS and Earth are close to each other; therefore, a deep network with more distinctive features may perform better. Evidently, Fig. 4.4 shows the deeper *VGG-19* model outperforms the other networks.

4.2 Real-time Saliency Extraction

In the previous sections, we have demonstrated using ConvNet semantic segmentation could extract the foreground spacecraft by pixel-wise classification. A major drawback of using CNN is the training needed to develop the network weights. The needs of large amounts of labeled training images and sufficient variety in these training images can be expensive to generate. Additionally, the training process itself requires careful adjustments to avoid early saturation and instability. To this end, we turn our attention to unsupervised

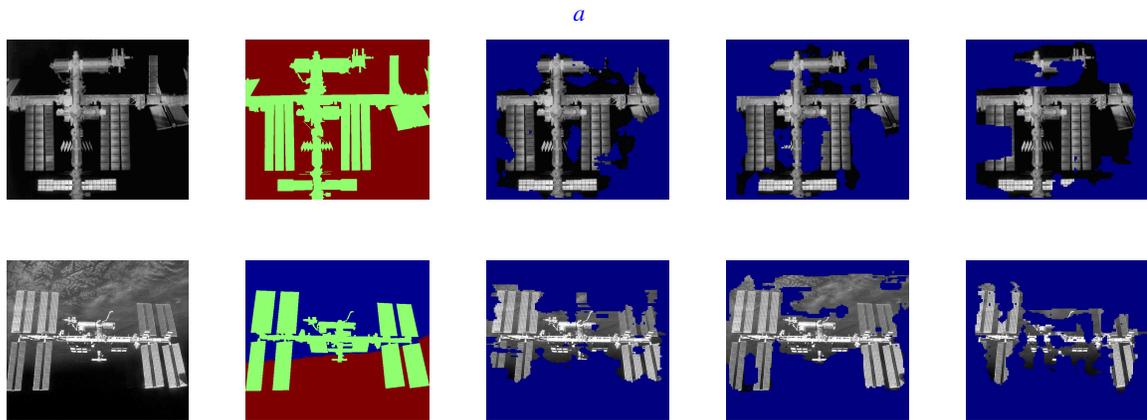


Figure 4.3: Semantic segmentation of the ISS. Top: IR camera image, bottom: photo camera image. Left to right: original, ground truth *AlexNet-8*, *UNet-8* and *VGG-19*.

^aGround Truth: https://github.com/ai-automata/170627_CNN_Segmentation

Table 4.3: Forward inference timing and semantic segmentation metric comparisons. Timing is an average of all evaluation images, semantic segmentation metric is based on the IR and the photo ISS images only.

Hdwr.	$\bar{t}(\text{ms})$	Network Model	Image	$\overline{\text{Acry}}$	$\overline{\text{Rcal}}$	Prec	$\overline{\text{Prec}}$	\bar{J}	J_{fw}
CPU	11.80	<i>AlexNet-8</i>	IR	0.68	0.67	0.71	0.73	0.50	0.50
	37.15	<i>UNet-8</i>		0.78	0.75	0.82	0.82	0.63	0.63
	61.93	<i>VGG-19</i>		0.59	0.58	0.61	0.64	0.40	0.40
GPU-1	0.58	<i>AlexNet-8</i>	Photo	0.68	0.58	0.53	0.50	0.33	0.49
	0.77	<i>UNet-8</i>		0.72	0.64	0.59	0.71	0.38	0.56
	1.13	<i>VGG-19</i>		0.63	0.48	0.44	0.41	0.25	0.38

a

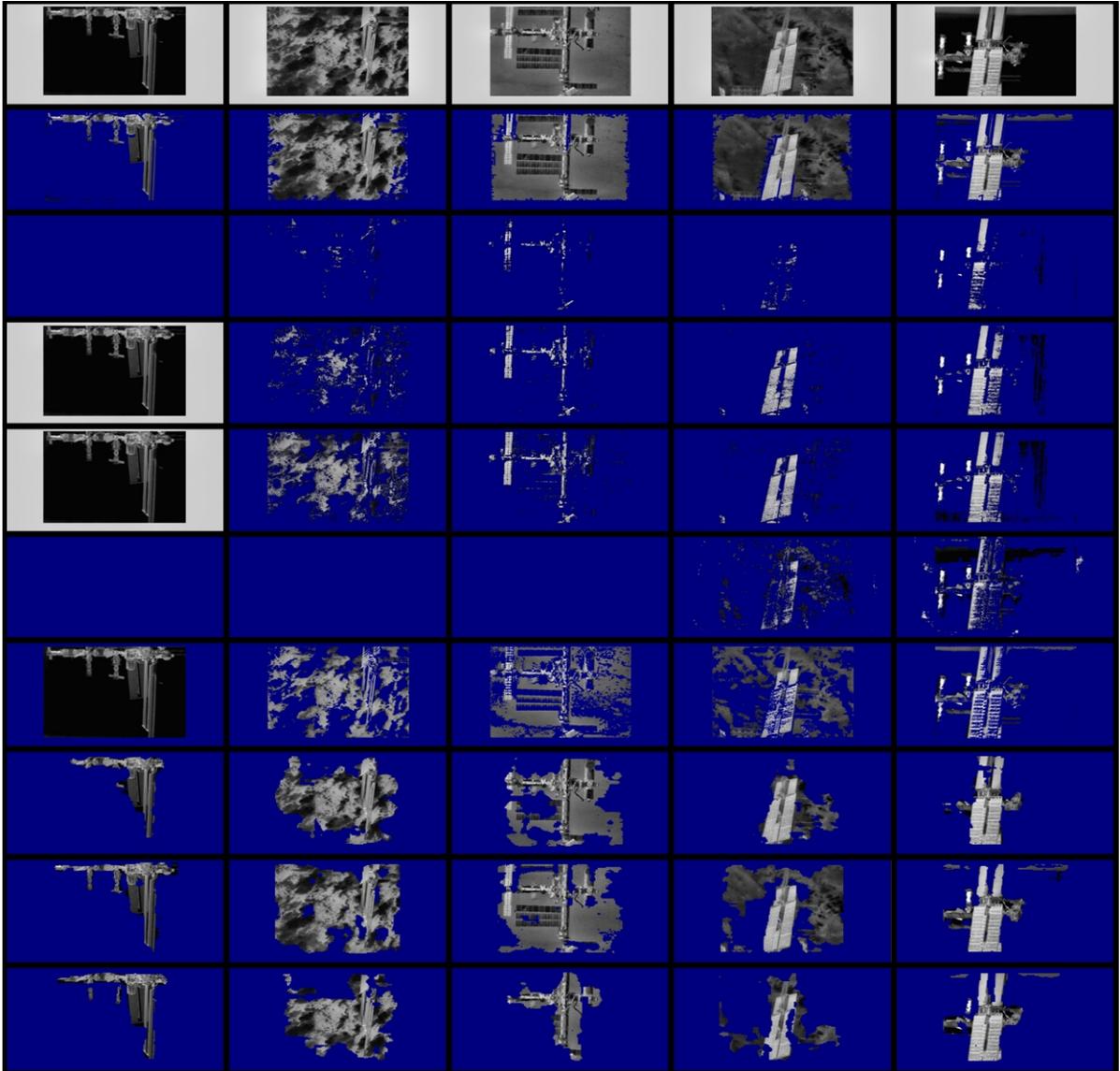


Figure 4.4: Background subtraction and semantic segmentation results, images are taken from STS-135 mission undocking sequence with background Earth motion. Equal partitioned frames are selected: 1, 55, 110, 164, and 219. Method as follows: level 1-original , level 2-manual , level 3-MOG [5] , level 4-MOG2 [6] , level 5-KNN [7] , level 6-GMG [8] , level 7-ADP [9] , level 8-AlexNet-8 , level 9-UNet-8 , level 10-VGG-19. MOG, MOG2, KNN, and ADP methods requires initialisation on the first frame, GMG method requires 120 frames to initialise.

^ahttps://youtu.be/B9ehT1Di_2w

methods that do not require complex pre-inference training and can run in real-time. The image saliency generation approach is an excellent candidate that would satisfy both requirements. Chapter 1 Sec. 1.4.2 provides background into image saliency generation. We focus on bottom-up image driven methods that do not require labeled training data.

While the bottom-up saliency detection methods are numerous, our interest is on those that are real-time capable. Furthermore, the attention models such as those by Itti [84], Hou [180] and Seo [190] computes the locations for where to direct visual attention but do not provide precise region boundaries of the foreground object. Our approach is to invoke methods that have sufficient precision in generating saliency maps of the desired foreground and is fast enough to be implemented in real-time. Three of the latest methods that stands out in this regard is Regional Contrast (RC) [167], Minimum Barrier Distance (MBD) [185] and Graph Manifold Ranking (GMR) [99,373]. Regional Contrast was proposed by Cheng *et al.* [167] to extract saliency from local regions as the weighted sum of colour contrast. The RC method uses a graph-based [374] image segmentation to first generate the working regions then creates a colour histogram and applies a histogram smoothing technique to reduce noise. Cheng [167] also uses RC to supplement *SaliencyCut*, a *GrabCut* [375] based segmentation in producing high-quality foreground masks. Our testing show RC is highly efficient to compute, with the most expensive module being the initial region segmentation. Zhang *et al.* [185] proposed a fast raster-scanning algorithm to approximate the MBD transform [376]. A central idea in the MBD is the *Image Boundary Connectivity Cue* which assumes background regions are connected to the image border, a variation of centre-surround [84]. While more precise than its geodesic counterpart, MBD by itself lacks the desired accuracy when compared to RC. An extended version, MB+, was proposed by adding *Image Boundary Contrast* (IBC) map using border pixels as colour contrast seeds in the whitened colour space. Unfortunately, the proposed IBC Map is computed by using individual pixels; as our testing show this can be an expensive addition to the MBD. Finally, Yang *et al.* [99] proposed GMR saliency using document ranking; GMR solves an optimisation problem by defining a graph-based ranking cost function. The minimum solution is an OAM based on weights of colour distances in *CIE Lab* colour space. GMR is elegant and fast; however, it requires a matrix inverse that can be time-consuming

and unstable when there is a large number of superpixels. To this end, one of our contributions is to enhance the inverse process by using PCA approximations. We present three algorithms based on the GMR model and improve its speed and precision.

4.2.1 Overview

We begin by comparing the timing of traditional and state-of-the-art methods from various reported studies discussed in Chapter 1 Sec. 1.4.2 and in Sec. 4.2 of this chapter. Models are discarded if they cannot be practically implemented in real-time even if they can produce highly precise saliency maps. Out of the remaining saliency model candidates, we found GMR [99], MBD [185] and RC [167] to have the best potential for highly precise real-time operations. We evolve the GMR method to meet our timing requirement of 50 ms per frame while still maintaining and improving its precision. Our performance space is bounded by the mean F -measure, the maximum AUC, and the minimum computation time. Instead of developing one approach that dilutes the conflicting requirements, we developed three algorithms that maximise each performance direction. We name these models precision (*prc*), fast (*fst*), and fast maximum-precision (*fmx*). We achieved our timing requirement with the *fst* and *fmx* models. Our *fmx* model has top precision versus recall curves while our *fst* model is the fastest design with competitive precision. We also achieve the highest mean F -measure overall by using our *prc* model. While the *prc* model did not meet our timing requirement, it is still 3 times faster than RC and GMR methods. All developed methods are based in GMR. The *prc* model includes a novel weighted orientation histogram feature that was the fastest compared to other non-colour based image descriptors which improves the saliency generation of the monochromatic image. Both *fmx* and *fst* models have inherited elements from the MBD and RC techniques. Details of each model are provided in Sec. 4.2.6.

4.2.2 Speed Improvements

The highest computation cost to GMR [99] results from the image size and the number of graph nodes. Our approach is to reduce the number of nodes, add stability to the GMR OAM computation, and at the same time increase the distinctiveness between foreground and background ranking scores especially when the colour information is not available.

Superpixels

Superpixels are used to identify locations and boundaries of graph nodes. They reduce the pixel-wise operations by several hundred times and are essential in the design of a real-time algorithm for saliency detection. Three popular superpixel techniques are Linear Spectral Clustering (LSC) [198], Simple Linear Iterative Clustering (SLIC) [197], and Superpixels Extracted via Energy-Driven Sampling (SEEDS) [199]. An example superpixelation of an ISS infrared image is shown in Figure 4.5 for the three described methods. LSC uses normalised cuts with colour similarity and space proximity [198]; unfortunately, LSC contains undesirable local discontinuities such as those shown in Figure 4.5(a). The LSC computation time was 7.941 ± 0.869 ms for the ISS test image with roughly 70 superpixels discounting the discontinuous subpixels. SLIC is a graph based algorithm treating each pixel as a graph node [197], similar to LSC, the number of pixels can be consistently specified and the pixels are similar in size. SLIC takes longer time to compute than both LSC and SEEDS but it generates the most stable and consistent superpixel map. For the ISS test image, the SLIC method timing was 11.610 ± 0.071 ms. To our knowledge, all GMR related saliency models uses SLIC to superpixelate the input image. Unlike the SLIC which grows superpixels by clustering around centres, SEEDS starts from a grid partitioning and iteratively refines the superpixel boundaries [199]. Our tests show that SEEDS is the fastest superpixel method at 6.078 ± 0.028 ms for the ISS image. However, the SEEDS superpixels are not consistent in size and the shape and quantity cannot be directly controlled. Figure 4.6 compares timing and precision performance for the various superpixel methods. LSC was the slowest and least precise overall. SLIC and SEEDS have roughly the same precision performance. SLIC is faster when the number of superpixels is low, but performs slower when the number of superpixels is higher. SLIC is used in the *prc* and the *fst* model, while SEEDS is used in the *fmX* model.

4.2.3 Graph Manifold Ranking

Manifold Ranking or GMR is a rating algorithm that spreads the seeding query scores to neighbouring nodes via the weighted network [377]. GMR has been widely adopted for document [378] and image retrievals [379]. The standard GMR framework is provided as follows: given an image with N number of superpixels, each superpixel is considered

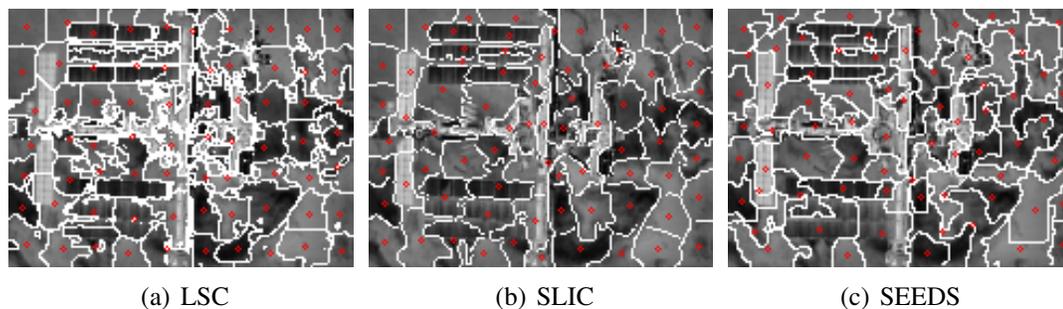


Figure 4.5: Superpixel breakdown of an infrared ISS image. Subfigures 4.5(a) to 4.5(c) shows LSC, SLIC, and SEEDS methods respectively. Red circles indicate locations of the superpixel spatial centroid.

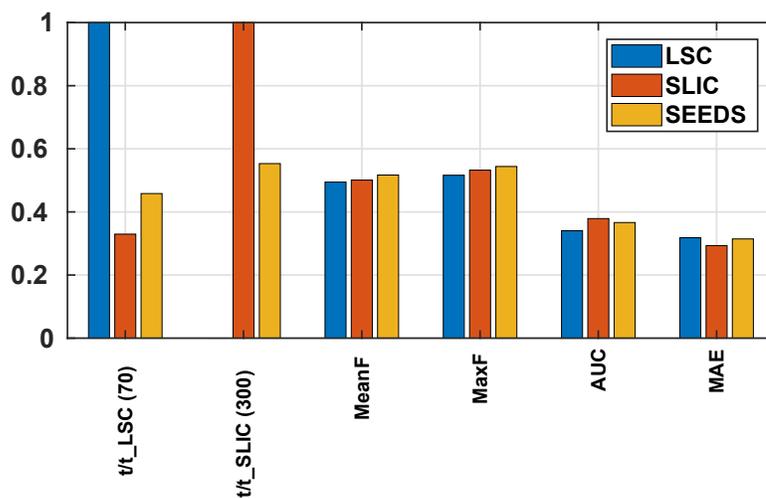


Figure 4.6: Superpixel performance comparisons. The first two columns indicate timing ratios dividing LSC and SLIC timing for 70 and 300 superpixels respectively. Columns 3 to 6 provides Mean F -measure, Max F -measure, AUC, and MAE.

as regions with some given feature vector $\mathbf{h}^{(i)} \in \mathbb{R}^m$. The set of feature vectors for all regions is $\bar{\mathbf{V}} = \{\mathbf{h}^{(1)}, \dots, \mathbf{h}^{(N)}\}^{m \times N}$. We select a subset of $\bar{\mathbf{V}}$ as seeds and rank the rest of the set base on their relevance to the seeding queries. Let us define $\mathbf{f} : \bar{\mathbf{V}} \rightarrow \mathbb{R}^N$ denote the ranking function for each region in N , such that $\mathbf{f} = [f^{(1)}, \dots, f^{(N)}]^T$. Let define $\mathbf{y} = [y^{(1)}, \dots, y^{(N)}]^T$ as an indication vector where 1 is to perform a query on $\mathbf{h}^{(i)}$ and 0 otherwise. Then define a graph $G = (\bar{\mathbf{V}}, \bar{\mathbf{E}})$ over the image regions, where the nodes $\bar{\mathbf{V}}$ are region features and $\bar{\mathbf{E}} = \mathcal{E}(\mathbf{x})$ are the edges which are weighted by the *affinity matrix* $\mathbf{W} = [w^{(ij)}]$ where $\mathbf{W} \in \mathbb{R}^{N \times N}$. The *degree matrix* $\mathbf{D} = \text{diag}\{d^{(11)}, \dots, d^{(NN)}\}$ is computed by $d^{(ii)} = \sum_{j \in N} w^{(ij)}$. The optimal ranking is computed by using the optimisation cost function,

$$\mathbf{f}^* = \underset{\mathbf{f}}{\operatorname{argmin}} \frac{1}{2} \left(\begin{array}{l} \sum_{i,j=1}^N w^{(ij)} \left\| \frac{f^{(i)}}{\sqrt{d^{(ii)}}} - \frac{f^{(j)}}{\sqrt{d^{(jj)}}} \right\|^2 \\ + \left(\frac{1}{\alpha} - 1 \right) \sum_{i=1}^N \|f^{(i)} - y^{(i)}\|^2 \end{array} \right), \quad (4.1)$$

where α is optimally tuned to 0.99 and 0.9 for the standard datasets and the SatSeg dataset respectively. For any given superpixel, its adjacent neighbours and the close-loop border boundaries are used for the feature distance differencing. Setting the derivative of Eq.(4.1) to zero, the resulting ranking function is

$$\mathbf{f}^* = \bar{\mathbf{A}}\mathbf{y} = \left(\mathbf{1} - \alpha \mathbf{D}^{-\frac{1}{2}} \mathbf{W} \mathbf{D}^{-\frac{1}{2}} \right)^{-1} \mathbf{y}, \quad (4.2)$$

where $\mathbf{1}$ is the identity matrix and $\bar{\mathbf{A}}$ is the normalised OAM. Yang *et al.* [99] proposed to use the unnormalised OAM $(\mathbf{D} - \alpha \mathbf{W})^{-1}$ for better performance. For background seeding queries, we take the normalised complementary vector for the foreground ranking score

$$\bar{\mathbf{f}}^* = \mathbf{1} - \frac{\mathbf{f}^*}{\underset{f^*}{\operatorname{argmax}} f^{*(i)}}, \quad (4.3)$$

where $f^{*(i)}$ is the ranking score at the i -th node. This ranking score is used to label individual superpixels in the final saliency map. The weighting is computed as

$$w^{(ij)} = \exp\left(\frac{\check{D} - D^{(ij)}}{\delta(\hat{D} - \check{D})}\right), \quad (4.4)$$

where $i, j \in \hat{\mathbf{V}}$, δ is optimally tuned to 0.1. We use the L1-norm instead of the L2-norm for $D^{(ij)}$ [191],

$$D^{(ij)} = \sum_{k=1}^m |h_k^{(i)} - h_k^{(j)}|, \quad (4.5)$$

where the maximum and minimum feature vector difference is $\hat{D} = \arg \max_{i, j \in \hat{\mathbf{V}}} D^{(ij)}$, and $\check{D} = \arg \min_{i, j \in \hat{\mathbf{V}}} D^{(ij)}$. While the change to L1-norm did not result in increased precision, it is slightly faster than the L2-norm implementation. We also compared the L1 and L2-norm in whitened feature space. Results show the difference in mean ranking scores for the foreground and background computed using the ground truth masks increased significantly; however, the variance of these ranking scores are still not concentrated enough to translate into higher overall precision performance.

The background seeding taken from each of the outer border superpixels is ranked by Eq.(4.2) and Eq.(4.3) to form the foreground saliency maps. The foreground maps based on the four borders are piece-wise multiplied

$$\bar{\mathbf{f}}_f^* = \bar{\mathbf{f}}_t^* \circ \bar{\mathbf{f}}_b^* \circ \bar{\mathbf{f}}_l^* \circ \bar{\mathbf{f}}_r^*, \quad (4.6)$$

where the subscripts f, t, b, l, r denotes foreground, top, bottom, left, and right respectively. A mean value binary threshold is applied to $\bar{\mathbf{f}}_f^*$. The resulting normalised ranking scores are fed into Eq.(4.2) again for the final foreground saliency map. The ranking scores are transformed back to the saliency image space by $S_f^{(i)} = \bar{f}^{*(i)}$ for $i = \{1 \dots N\}$ where i is the superpixel label index.

Optimal Affinity Matrix from PCA Inversion

The *learnt optimal affinity matrix* \mathbf{A} is of the size of $N \times N$. For a low number of superpixels (*i.e.* $N \leq 200$), the inversion can be computed quickly. However, as the number of

superpixels increases, \mathbf{A} takes much longer to compute and has more chances of becoming unstable. For general application of GMR for document and image retrieval, attempts have been made to reduce the dimension of \mathbf{A} by efficient computations via sparse matrices [380] and by designing the weighting matrix as a separable symmetrical matrix such that $\mathbf{W} = \mathbf{Z}^T \mathbf{Z}$ [379]. In our GMR framework, however, the weighting matrix cannot be decomposed into a single \mathbf{Z} matrix. We derive a new form for representing the non-normalised OAM using PCA approximation for matrix reduction.

First, we decompose the weighting matrix by *Singular Value Decomposition*,

$$\mathbf{W} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T = \mathbf{P}\mathbf{Q}, \quad (4.7)$$

where $\mathbf{U} = \mathbf{P}$ and contains the columns of eigenvectors. The diagonal of $\mathbf{\Sigma}$ contains the square root of the eigenvalues $\sqrt{\lambda^{(i)}}$, $i = \{1 \dots N\}$ descending from large to small. Let us define $\mathbf{E} = \mathbf{D}^{-\frac{1}{2}}\mathbf{P}$ and $\mathbf{F} = \mathbf{Q}\mathbf{D}^{-\frac{1}{2}}$, then the normalised OAM can be written as $\bar{\mathbf{A}} = (\mathbf{1} - \alpha\mathbf{E}\mathbf{F})^{-1}$.

Lemma 4.2.1. Given the matrix $\bar{\mathbf{A}} \in \mathbb{R}^{N \times N}$, $\mathbf{E} \in \mathbb{R}^{N \times d}$, $\mathbf{F} \in \mathbb{R}^{d \times N}$, and $\bar{\mathbf{A}} = (\mathbf{1} - \alpha\mathbf{E}\mathbf{F})^{-1}$ such that the inverse exists, then an alternative form for $\bar{\mathbf{A}}$ can be written as $\bar{\mathbf{A}} = \mathbf{1} - \mathbf{E}(\mathbf{F}\mathbf{E} - \frac{1}{\alpha}\mathbf{1}_d)^{-1}\mathbf{F}$, where $\mathbf{1}_d \in \mathbb{R}^{d \times d}$ is the identity matrix and $d \leq N$.

Proof. If the alternative form of $\bar{\mathbf{A}}$ from Lemma 4.2.1 is equivalent, then when multiplied by the inverse it will result in the identity.

$$\begin{aligned} & (\mathbf{1} - \alpha\mathbf{E}\mathbf{F}) \left(\mathbf{1} - \mathbf{E} \left(\mathbf{F}\mathbf{E} - \frac{1}{\alpha}\mathbf{1}_d \right)^{-1} \mathbf{F} \right) \\ &= \mathbf{1} - \alpha\mathbf{E}\mathbf{F} - \mathbf{E} \left(\mathbf{F}\mathbf{E} - \frac{1}{\alpha}\mathbf{1}_d \right)^{-1} \mathbf{F} + \alpha\mathbf{E}\mathbf{F}\mathbf{E} \left(\mathbf{F}\mathbf{E} - \frac{1}{\alpha}\mathbf{1}_d \right)^{-1} \mathbf{F} \\ &= \mathbf{1} - \alpha\mathbf{E}\mathbf{F} + \alpha\mathbf{E} \left(-\frac{1}{\alpha}\mathbf{1}_d + \mathbf{F}\mathbf{E} \right) \left(\mathbf{F}\mathbf{E} - \frac{1}{\alpha}\mathbf{1}_d \right)^{-1} \mathbf{F} \\ &= \mathbf{1} \end{aligned} \quad \blacksquare$$

We take an additional step to develop the equivalent inverse equation for the non-normalised OAM where

$$\begin{aligned}
\mathbf{A} &= \mathbf{D}^{-\frac{1}{2}} \bar{\mathbf{A}} \mathbf{D}^{-\frac{1}{2}} \\
&= \left(\mathbf{D}^{\frac{1}{2}} \mathbf{D}^{\frac{1}{2}} - \alpha \mathbf{D}^{\frac{1}{2}} \mathbf{D}^{-\frac{1}{2}} \mathbf{W} \mathbf{D}^{-\frac{1}{2}} \mathbf{D}^{\frac{1}{2}} \right)^{-1} \\
&= (\mathbf{D} - \alpha \mathbf{P} \mathbf{Q})^{-1}.
\end{aligned} \tag{4.8}$$

The inverse form can be developed as the following,

Corollary 4.2.2. Given the matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$, $\mathbf{P} \in \mathbb{R}^{N \times d}$, $\mathbf{Q} \in \mathbb{R}^{d \times N}$, and $\mathbf{A} = (\mathbf{D} - \alpha \mathbf{P} \mathbf{Q})^{-1}$ such that the inverse exists, then an alternative form for \mathbf{A} can be written as

$$\mathbf{A} = \mathbf{D}^{-1} \left(\mathbf{1} - \mathbf{P} \left(\mathbf{Q} \mathbf{D}^{-1} \mathbf{P} - \frac{1}{\alpha} \mathbf{1}_d \right)^{-1} \mathbf{Q} \mathbf{D}^{-1} \right), \tag{4.9}$$

where $\mathbf{1}_d \in \mathbb{R}^{d \times d}$ is the identity matrix and $d \leq N$.

Proof. Substituting Lemma 4.2.1 into Eq.(4.9), and using the previous definitions for \mathbf{E} and \mathbf{F} ,

$$\begin{aligned}
\mathbf{A} &= (\mathbf{D} - \alpha \mathbf{W})^{-1} \\
&= \mathbf{D}^{-\frac{1}{2}} \left(\mathbf{1} - \mathbf{E} \left(\mathbf{F} \mathbf{E} - \frac{1}{\alpha} \mathbf{1}_d \right)^{-1} \mathbf{F} \right) \mathbf{D}^{-\frac{1}{2}} \quad \blacksquare \\
&= \mathbf{D}^{-1} \left(\mathbf{1} - \mathbf{P} \left(\mathbf{Q} \mathbf{D}^{-1} \mathbf{P} - \frac{1}{\alpha} \mathbf{1}_d \right)^{-1} \mathbf{Q} \mathbf{D}^{-1} \right)
\end{aligned}$$

Using the results of Corollary 4.2.2, the OAM may be approximated by taking the most relevant Eigen components effectively reducing the inverse dimension from N to a lower value d . We denote the approximate OAM, degree matrix and decomposition of the weighting matrix using the tilde notation as $\tilde{\mathbf{A}}$, $\tilde{\mathbf{D}}$, $\tilde{\mathbf{P}}$ and $\tilde{\mathbf{Q}}$. We can adaptively compute the number of most relevant Eigen components to keep by setting some threshold to the sum of the full Eigen components by the following,

$$\frac{\sum_{i=1}^d |\sqrt{\lambda^{(i)}}|}{\sum_{i=1}^N |\sqrt{\lambda^{(i)}}|} \geq Thr, \tag{4.10}$$

where 1 to d is the reduced number of components in maximum descending order, the values for $\sqrt{\lambda^{(i)}}$ may be extracted from Σ of Eq.(4.7). PCA ensures an optimally minimised OAM approximation; we may manually set the Thr tolerance based on timing and precision requirements. Figure 4.7 provides results for the various threshold levels varied from 0.6 to 1. We observe when $Thr = 1$ the reduced method is slower due to the additional steps in the modified form. Time-savings increase with reduced thresholds and increasing number of superpixels. By contrast, the error in the OAM increases with threshold reduction. Interestingly, a minimum point is observed at 168 pixels resulting from the balance between the omitted components and the retained ones. To take a broader view of the OAM reduction effectiveness, we define the distinctiveness as the difference between foreground and background ranking scores using the ground truth mask's intersection with the final f^* . The distinctiveness from both Figure 4.7(c) and 4.7(d) shows a steady reduction in GMR performance as OAM reduction increases for colour and grayscale images respectively.

High-frequency Response Attention Driven Seeding

The final GMR ranking computed by Eq.(4.2) is a function of the OAM A and the estimated nodes of the foreground y as queries. We replaced the estimated foreground query with the ground truth foreground nodes and found significant improvements in the overall precision. This indicates GMR precision is sensitive to the quality of the seeds used in the ranking optimisation. We developed a high-frequency response attention driven seeding scheme to improve background estimation from the surrounding borders. Using the border nodes for background seeding has been used by many models [99, 185, 203, 206, 381]. We remove the border seeds from the background query if there is evidence these are foreground nodes and replace them with the inner-layer neighbour background estimate. The target object in the spacecraft application typically contains strong artificial lines and edges compared to the softly blended background scene. Therefore we extract these high-frequency lines by applying a 3×3 Laplacian filter, $\mathcal{L}(x)$, on the grayscale input image I and blurred by a square box filter, $\mathcal{B}(x)$, of the size K_B , to remove noise. We focus the foreground region from eye fixation attention cues computed by SR [180]. Our testing show SR to be faster and produce better results than using QFT. The attention cue is combined with the high-frequency region by using a Gaussian distribution map, $\mathcal{G}(x)$, centred at the SR moment

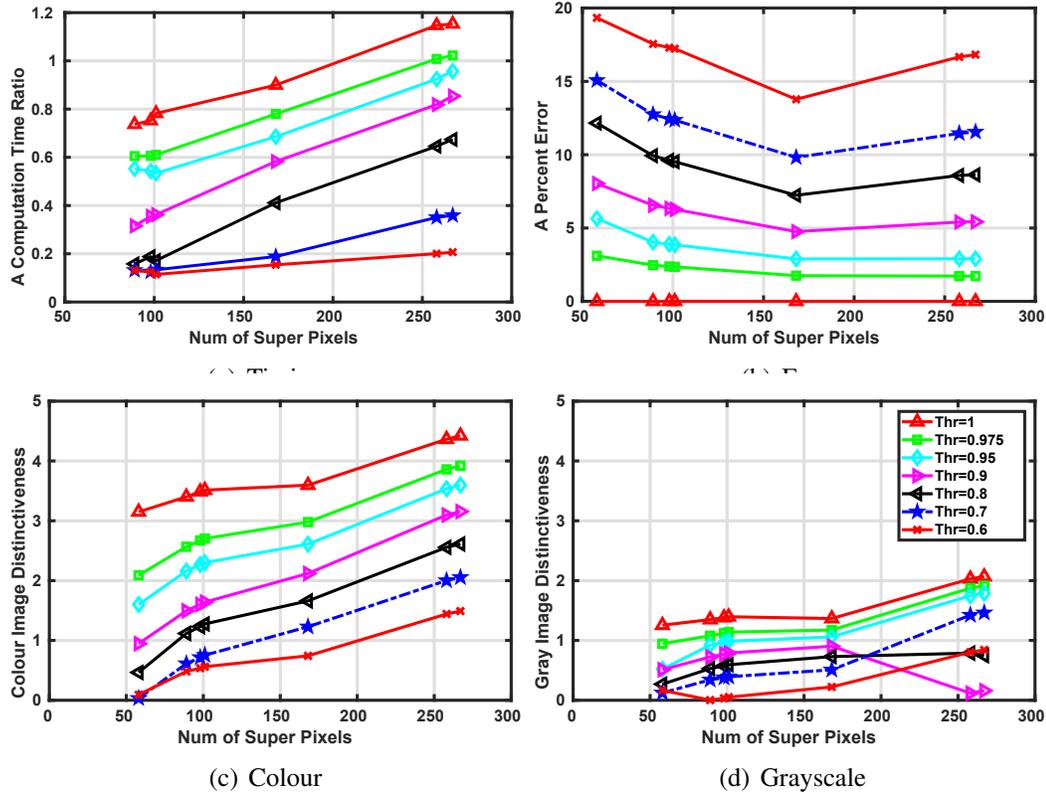


Figure 4.7: OAM reduction results. Each data point is the average of all 32,536 image calculations. Threshold level corresponds to Eq.(4.10). Figure 4.7(a) shows the timing ratio between reduced OAM over non-reduced for a single OAM calculation. Figure 4.7(b) shows the average-relative-percent-error in the reduced OAM. Subfigure 4.7(c) and 4.7(d) shows the distinctive difference between the average foreground and background ranking score for all colour and grayscale images respectively.

centroid, $\mathcal{MC}(SR(\mathbf{I}))$. We then take the intersection between the estimated foreground responses with the border seeds to replace any lost nodes with the estimated background node neighbours. The seeding process is depicted in Figure 4.8.

4.2.4 Monochromatic Features

Colour features are essential in distinguishing objects from its background; we use colour features in the *CIE Lab* colour space similar to many other methods [99, 167]. Our testing confirms higher performance using *Lab* colour space than *RGB*. For efficiency, we compute the colour space conversion from *RGB* to *Lab*, $\mathbf{I}_{Lab} \leftarrow \mathcal{LAB}(\mathbf{I})$, only once for SLIC super-pixelation and in our modified GMR calculation. Unfortunately, in the spacecraft GNC

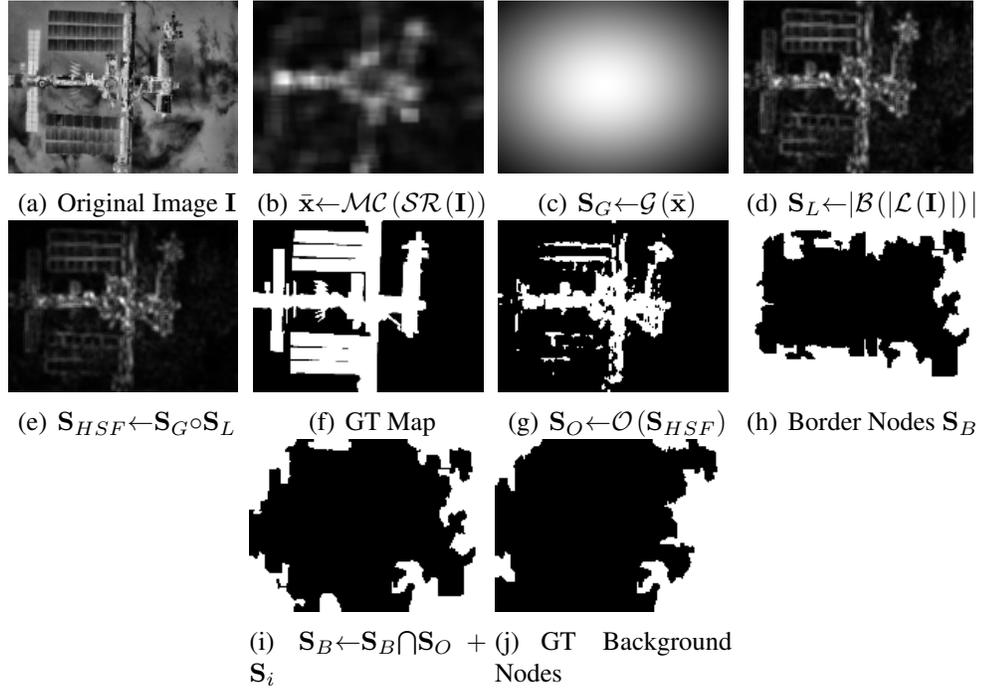


Figure 4.8: Algorithm sequence for border seeds combined with the foreground estimate. The input image I in 4.8(a) is passed into SR in 4.8(b), where the SR moment centroid is used to centre the Gaussian centredness map S_G in 4.8(c). A high-frequency response S_L is computed by the 3×3 Laplacian and the $K_B \times K_B$ box filter in 4.8(d). The final foreground estimate based on image frequency is S_{HSF} in 4.8(e) which can be compared with the ground truth masks in 4.8(f). S_{HSF} is threshold using the OTSU method in 4.8(g), and then combined with the border nodes S_B from 4.8(h), for a better estimation of the background border seeds in 4.8(i), where S_i are the replacement background nodes. S_B can be compared with the ground truth border seeds in 4.8(j).

application, data rate and memory restrictions require the use of low-resolution monochromatic images, and only grayscale images are available from infrared imagers. To this end, we focus on the development of an alternative real-time solution using image orientation and texture instead of colour.

Weighted Histogram of Orientation

Local image orientation also provides information between various objects in the scene. We are inspired by Jung's [196] approach in using an orientation histogram from local image gradients. Instead of finding the orientation differences from the dominant direction, we use the orientation histogram directly as the image feature. We built our orientation histogram

weighted by the gradient magnitude; we call this approach the Weighted Histogram of Orientation (WHO) feature since the histogram encodes both the local direction and strength of the image gradient. Let an 8-bit grayscale local patch be denoted by \mathbf{I} and the patch centre is located at the superpixel spatial centroid. The first-order image intensity derivative for an individual pixel in the local patch is $I_x^{(ij)} = \partial I^{(ij)} / \partial x$ and $I_y^{(ij)} = \partial I^{(ij)} / \partial y$, where i and j denotes the row and column index of the local patch. We compute the gradient image by convolving \mathbf{I} with the 3×3 Sobel kernel. From the patch gradients, the orientation and magnitude may be computed as $\theta^{(ij)} = \arctan\left(I_y^{(ij)} / I_x^{(ij)}\right)$ and $\xi^{(ij)} = \|[I_x^{(ij)}, I_y^{(ij)}]\|_2$. We use the *atan2* function which limits the angle range between $[-\pi, \pi]$; this may result in two points under the same orientation line image to have two opposite direction angles. To remove these opposite angles that could hinder the distance comparison, we added π to all negative angles, effectively changing the orientation limits between $[0, \pi]$. We then collected the orientation of every pixel in the local patch by adding $\xi^{(ij)}$ to the histogram bin. For each local patch, we normalise the WHO histogram between $[0, 1]$; this allows better intra-node comparisons and avoids possible extreme values from border nodes to overwhelm the histogram magnitude. For colour images, we may keep the WHO features for each image channel.

For each graph node, we have features from the *Lab* colour vector and the WHO feature vector having the dimensions of the specified orientation bins. The *a* and *b* channels are by definition zero when the image is monochromatic. The feature difference as noted by Eq.(4.5) is used for the colour vector, while the similarity distance of the WHO feature is compared using the *Bhattacharyya* distance between histograms. Since the histogram distance and the intensity distance may have very different magnitudes, we use the following equation to bring the two distances into the same range,

$$D = D_c + \bar{D}_c(1 + D_w), \quad (4.11)$$

where D_c is the L1-norm of colour intensity as given by Eq.(4.5), \bar{D}_c is the average of all D_c computed recursively, D_w is the intra-node histogram distance. Equation 4.11 allows the colour or orientation histogram to be independent when either one could have the potential to be zero locally.

We compute a range of local patch size and the number of bins for the WHO feature,

and the results are in Figure 4.9. We observe a 17 percent timing increase after introducing the WHO feature with the smallest aperture of 3×3 and bin width of 30 degrees. The timing increased 46 percent when the patch aperture is 21×21 with a bin width of 5 degrees. However, the increased computation time did not translate into better precision performance. In fact, the performance is lower with the highest aperture and finest bin width. The optimal aperture bin width is 5×5 and 20 degrees respectively. Figure 4.9(b) shows using orientation in addition to the image intensity increased the mean F -measure for colour and grayscale images. Overall, there is a consistent increase in all performance metrics for grayscale images, but for colour images, the performance tends to reduce in maximum F -measure, AUC and MAE. This result is expected since the grayscale image lacks channel variation in the image intensity which is supplemented by the orientation histogram. The colour image already has good object distinctiveness, and its resolution could be diluted by the less precise orientation map. Based on these results, we implement the WHO feature only on grayscale images for optimal time and precision performance.

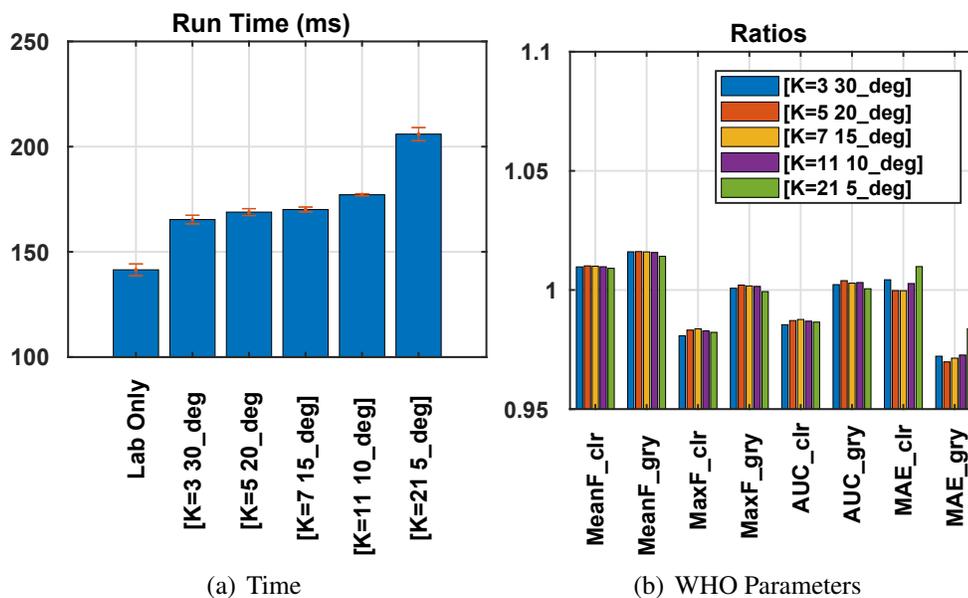


Figure 4.9: WHO parameter sensitivity study. Figure 4.9(a) provides timing for *Lab* colour-space-only and various $K \times K$ patch size and histogram bin delta angles. Error bars represent one standard deviation of the measured timing. Figure 4.9(b) provides ratio of the WHO parameter variation over the *Lab* colour-space-only metrics. The metrics are Mean F -measure, Max F -measure, AUC, and MAE for colour and grayscale images. Each data point is the average of all 32,536 images.

4.2.5 Image Texture from Feature Descriptors

Typical texture based approach uses Gabor [111], Laplacian of Gaussian and Gaussian filters to extract *textons* [112] from the local image. Our testing of the LM, S and MR8 filter banks [115] show this approach to be too computationally intensive for real-time application. We turned our focus to other image features for their description of the image texture from local difference in illumination. Image features, such as the SIFT descriptors [15], are designed for keypoint matching and SIFT has the property of indicating local intensity gradients. Two neighbouring nodes of the same foreground object under the same illumination should have closer local intensity gradient than with far away nodes of the background scene, and therefore in principle, image descriptors can also be used to describe image texture. During preliminary testing, we have discovered evidence of SIFT [15] and FREAK [11] image descriptors to be distinctive in the classification of the foreground and background regions. We formally evaluate the effectiveness of six image features using saliency metrics: AKAZE [10], FREAK [11], BRISK [12], BRIEF [13], SURF [14], and SIFT [15]. The descriptor distance can be computed as $D_f^{(ij)} = |\hat{\mathbf{h}}^{(i)} \cdot \hat{\mathbf{h}}^{(j)}|$ where $\hat{\mathbf{h}}$ is the L2 normalised unit feature vector. We add this feature distance to the colour space L1-norm distance the same way as in Eq.(4.11) by replacing D_w with D_f .

Results of the various image descriptor ratios over the colour-only baseline compared to the WHO feature ratio in timing and precision are provided in Fig. 4.10. Figure 4.10(b) shows all descriptors caused increases in the mean F -measure for colour and grayscale images, and grayscale images outperformed colour images for all metrics; this confirms the premise that descriptors can be used to add texture feature distinctiveness between nodes. When ordered by publication year, an increase in performance with newer feature descriptors can generally be observed; this again can be expected as the image descriptors are improved based on similar precision versus recall approach for matching nearest keypoint features. Finally, the WHO feature out-performed all image descriptors with the exception of colour image maximum F -measure, AUC and MAE against AKAZE. More importantly, the WHO feature is $3\times$ to $5\times$ faster than the conventional image descriptors as shown in Fig. 4.10(a). This result alone excludes the use of conventional image descriptors since they would violate our timing requirement. Based on these results, we confirm using WHO features for monochromatic image node representation in addition to the image intensity.

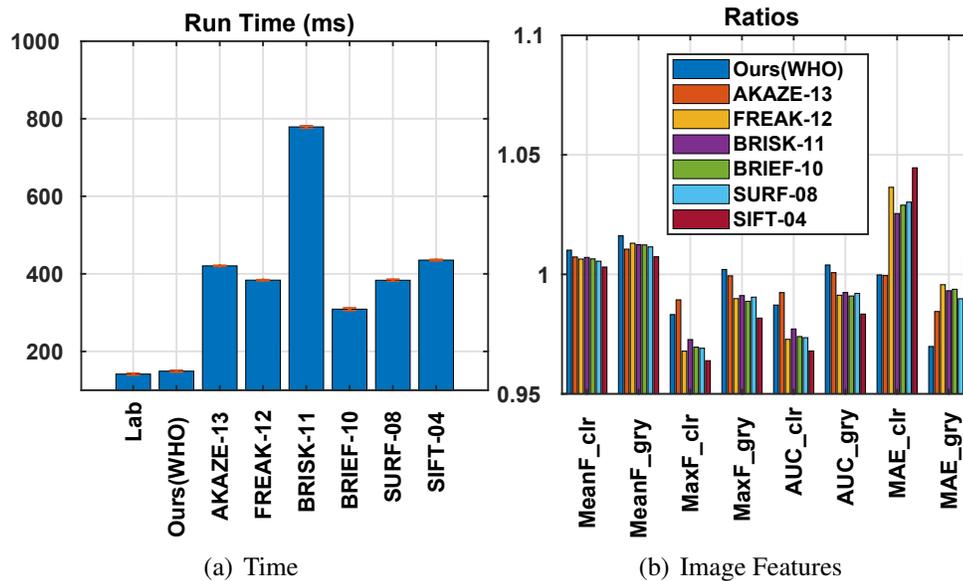


Figure 4.10: Image feature sensitivity study. Figure 4.10(a) provides timing comparisons for various image feature in addition to the *Lab* colour space. Figure 4.10(b) provides ratio of the various image features over the *Lab* performance. The same metrics and number of images are used as per descriptions in Fig. 4.9. The image feature cases are AKAZE [10], FREAK [11], BRISK [12], BRIEF [13], SURF [14], and SIFT [15]

4.2.6 Enhanced GMR

Our enhanced GMR method uses the best estimate foreground and background seeding rather than the traditional border seeding. We increased the speed of GMR by more than $11\times$ and improved its precision performance.

Precision Model

In the precision model (*prc*), we follow the GMR process described in Sec. 4.2.3 with the following modifications. First, we resize the input image to 160×120 and set the number of SLIC pixels to roughly 200. Computation speed is extremely sensitive to the number of pixels, a small increase in the number of superpixels may not change the precision performance, but it will make dramatic increases in computation time. We include the WHO features per Sec. 4.2.4 for grayscale images when computing the Laplacian weighting matrix. Sec. 4.2.5 shows the WHO feature is faster than all other tested image descriptors; however, the *Bhattacharyya* distance calculation is still ten times slower than the *Lab* L1-norm and can only be implemented in the *prc* model due to our timing requirement. The

weighting matrix is computed by an exhaustive neighbour search through the adjacent matrix for every node. Rather than computing the feature difference during the adjacent node search, we only mark the weighting matrix. The feature difference is then computed in all the marked weights, so there are no duplications in the difference calculation. This enhancement reduced the number of feature difference calculation from $\mathcal{O}(n^2)$ to $\mathcal{O}(n)$ where n is the number of adjacent intra-connections. Our timing test show this improvement reduced the overall computation time by two-third. Next, we implement PCA OAM reduction with a pixel limit of 150. If the full OAM cannot be inverted, we successively reduce the threshold tolerance by 95 percent for 5 iterations. In practice, the approximation of the OAM inversion is no longer corrupted after one reduction iteration. Finally, we apply contrast enhancement by a sigmoid function as described in Zhang *et al.* [185],

$$I^{(ij)} \leftarrow \frac{1}{1 + e^{-b(I^{(ij)} - 0.5)}}, \quad (4.12)$$

where b is optimally tuned to 10. The sigmoid contrast enhancement reduced the maximum F -measure, but significantly improved the mean F -measure and MAE.

Speed Optimised Model

Based on the findings from Sec. 4.2.3, we propose a speed optimised model (*fst*) that does not use the traditional border nodes as background seeding. Instead, we modify GMR, $\overline{\mathcal{GMR}}(\mathbf{x})$, to use an estimate of the foreground, \mathbf{S}_{fg} , and background seeds, $255 - \mathbf{S}_{fg}$, by combining principals of high frequency response, Gaussian centredness, MBD and modified RC. We modify RC, $\overline{\mathcal{RC}}(\mathbf{I}_{Lab})$, by removing the graph segmentation calculation which is the most costly module in the RC pipeline that is worth 27 percent of the entire RC process. Instead, we use the already computed superpixel map and compute the superpixel's spatial centroid. For the ISS test image with 70 superpixels, this reduced the region segmentation calculation from 45 ms to 12 and 6 ms for SLIC and SEEDS respectively. We propose a centredness map [185, 191] using a Gaussian distribution centred on the spatial moment centroid of the SR map. The Gaussian distribution has the following adaptive

scales,

$$\begin{bmatrix} \sigma_x \\ \sigma_y \end{bmatrix} = 4 \left(\begin{bmatrix} C \\ R \end{bmatrix} - 1 \right), \quad (4.13)$$

where R and C are the numbers of rows and columns of the input image. Let define $MT(\mathbf{x})$ as the mean binary threshold of an input map. The *fst* algorithm is provided in Algorithm 2.

Algorithm 2 The *fst* Algorithm

```

1: procedure FST(I)
2:   if Image larger than 120 rows then
3:     Resize image to 120 rows
4:   if Border frames exist then
5:     Remove border frames
6:   Check for grayscale image
7:    $\mathbf{I}_{Lab} \leftarrow \mathcal{LAB}(\mathbf{I})$ 
8:    $\mathbf{S}_S \leftarrow \mathcal{SLIC}(\mathbf{I}_{Lab}, N = 97)$ 
9:    $\bar{\mathbf{E}} \leftarrow \mathcal{E}(\mathbf{S}_S)$ 
10:   $\mathbf{S}_{SR} \leftarrow \mathcal{N}(\mathcal{SR}(\mathbf{I}, \sigma = 2))$ 
11:   $\mathbf{S}_G \leftarrow \mathcal{G}(\mathcal{MC}(\mathbf{S}_{SR}))$ 
12:   $\mathbf{S}_L \leftarrow \mathcal{N}(|\mathcal{B}(|\mathcal{L}(\mathbf{I})|, K_B = 5)|)$ 
13:   $\mathbf{S}_{HSF} \leftarrow \mathcal{N}(\mathbf{S}_{SR} + \mathbf{S}_L)$ 
14:   $\mathbf{S}_{HSF} \leftarrow \mathbf{S}_{HSF} + \text{Mean}(\mathbf{S}_{HSF})$ 
15:   $\mathbf{S}_{MBD} \leftarrow \mathcal{N}(\mathcal{MBD}(\mathbf{I}_{Lab}))$ 
16:   $\mathbf{S}_{RC} \leftarrow \overline{\mathcal{RC}}(\mathbf{I}_{Lab})$ 
17:   $\mathbf{S}_{fg} \leftarrow \mathcal{O}(\mathbf{S}_L) \cup \mathcal{MT}(\mathbf{S}_G \circ \mathbf{S}_{HSF} \circ \mathbf{S}_{MBD} \circ \mathbf{S}_{RC})$ 
18:   $\mathbf{S}_{fg} \leftarrow \mathcal{N}(\overline{\mathcal{GMR}}(\mathbf{S}_{fg}) \circ (255 - \overline{\mathcal{GMR}}(255 - \mathbf{S}_{fg})))$ 
19:  for  $i = 1 : \mathbf{S}_{fg}$  height do
20:    for  $j = 1 : \mathbf{S}_{fg}$  width do
21:       $\mathbf{S}_{fg}(i, j) \leftarrow \frac{1}{1 + e^{-b(\mathbf{S}_{fg}(i, j) - 0.5)}}$  where  $b = 10$ 
22:    Foreground Saliency Map:  $\mathbf{S}_{fg} \leftarrow \mathcal{N}(\mathbf{S}_{fg})$ 
23:    Foreground Mask:  $\mathbf{S}_{mask} \leftarrow \mathcal{M}(\mathbf{S}_{fg})$ 
24:  if Resized then
25:    Resize saliency map to original size

```

Fast Maximum Precision Model

The fast maximum precision model (*fm_x*) is optimised to provide maximum precision versus recall curve and AUC. Let define the function that extracts the top, bottom, left, and

right border nodes as $\mathcal{BDR}(x)$. The algorithm for fmX is provided in Algorithm 3.

Algorithm 3 The fmX Algorithm

```

1: procedure FMX(I)
2:   if Image larger than 120 rows then
3:     Resize image to 120 rows
4:   if Border frames exist then
5:     Remove border frames
6:   Check for grayscale image
7:    $\mathbf{S}_G \leftarrow \mathcal{G}(\mathcal{MC}(\mathcal{SR}(\mathbf{I}, \sigma = 8)))$ 
8:    $\mathbf{S}_L \leftarrow |\mathcal{B}(|\mathcal{L}(\mathbf{I})|, K_B = 3)|$ 
9:    $\mathbf{S}_{HSF} \leftarrow \mathbf{S}_G \circ \mathbf{S}_L$ 
10:   $\mathbf{I}_{Lab} \leftarrow \mathcal{LAB}(\mathbf{I})$ 
11:   $\mathbf{S}_S \leftarrow \mathcal{SEEDS}(\mathbf{I}, N = 225)$ 
12:   $\bar{\mathbf{E}} \leftarrow \mathcal{E}(\mathbf{S}_S)$ 
13:   $\mathbf{S}_O \leftarrow \mathcal{O}(\mathbf{S}_{HSF})$ 
14:  for  $i = top, bottom, left, right$  do
15:     $\mathbf{S}_B^{(i)} \leftarrow \mathcal{BDR}(\mathbf{S}_S, border = i) \cap \mathbf{S}_O + \mathbf{S}_i$ 
16:     $\mathbf{S}_{GMR} \leftarrow \overline{\mathcal{GMR}}(\mathbf{S}_B^t, \mathbf{S}_B^b, \mathbf{S}_B^l, \mathbf{S}_B^r)$ 
17:     $\mathbf{S}_{MBD} \leftarrow \overline{\mathcal{MBD}}(\mathbf{I}_{Lab})$ 
18:     $\mathbf{S}_{RC} \leftarrow \overline{\mathcal{RC}}(\mathbf{I}_{Lab})$ 
19:     $\mathbf{S}_{fg} \leftarrow \mathbf{S}_G \circ \mathbf{S}_{GMR} \circ \mathbf{S}_{MBD} \circ \mathbf{S}_{RC}$ 
20:  if Resized then
21:    Resize saliency map to original size

```

4.2.7 Dataset and Metrics

Standard Datasets

While our design focuses on spacecraft applications, our developed models can also be used for general images. To objectively evaluate against the benchmark saliency models, we use several standard datasets for colour and grayscale images. The descriptions for the standard datasets is provided in Tables 4.4.

Satellite Segmentation Dataset

Our primary goal is to develop a real-time saliency detection model for the spacecraft GNC pipeline. While there are many image datasets available for saliency and segmentation

Table 4.4: Image datasets. NOTE:starred references (*) are where the datasets were downloaded from.

Name	Ref.	Size
ECSSD	[299]	1,000
DUT-OMRON	[99]	5,168
MSRA10K	[169] [167]*	10,000

Table 4.5: SatSeg Description

Colour	Grayscale	Description
2	9	Control images that are non-nadir pointing
10	20	Colour and infrared images of the ISS with Earth background
37	3	Colour images of various spacecraft and space debris
5	14	Colour and infrared images of laboratory RSM

studies, to our knowledge there is no image dataset developed explicitly for spacecraft applications. We generate a Satellite Segmentation (SatSeg) dataset compose of 100 colour and grayscale spacecraft images captured by photo and infrared cameras that are representative of the various mission scenarios ranging from flight and laboratory tests. Ground truth data was produced manually as foreground(255) and background(0) identification masks. Table 4.5 provides a detail accounting of the *SatSeg* images. We make the SatSeg dataset freely available to download from our project website for future developments by the imaging community.^a

Model Comparisons

We compared our models with 18 traditional and state-of-the-art saliency detection and segmentation models provided in Table 4.6. These models were selected based on their real-time and near-real-time running performance. For ECSSD, DUT-OMRON and MSRA10K datasets, comparisons were made against LC [182], SR [180], QFT [181], AC [183], FT [186], MSS [189], HC [382], GMR [99], GC [192], GD [185], MBD [185], MB+ [185] and RC [167]. For the SatSeg tests, we added OTSU [139], WS [383], GCUT [375],

^a<http://ai-automata.ca/research/hisafe.html>

Table 4.6: List of benchmark models used for comparison. NOTE: starred references (*) is where source code or executable was downloaded from.

Code	Reference	Description
OTSU 75	[139]	OTSU Thresholding
WS 92	[383]	Watershed
GCUT 04	[375]	Grabcut
LC 06	[182] [167]*	Colour contrast
SR 07	[180]	Spectral Residual
QFT 08	[181]	Phase Spectrum Quaternion Fourier Transform
AC 08	[183] [155]*	Local contrast raster scan
FT 09	[186]	Frequency Tuned
MSS 10	[189]	Maximum Symmetric Surround
DRLSE 10	[285]	Distance Regularized Level-Set Evolution
HC 11	[382]	Histogram-based Contrast
GMR 13	[99]	Graph Manifold Ranking
GC 13	[192]	Global Cues
BING 14	[384]	BInarized Normal Gradients
GD 15	[185]	Geodesic
MBD 15	[185]	Minimum Barrier Distance
MB+ 15	[185]	MBD Extended
RC 15	[167]	Regional Contrast

DRLSE [285] and BING [384] to the previous list, this allowed running-time comparisons between saliency generation, direct thresholding, segmentation and objectiveness methods. WS, GCUT, DRLSE are segmentation methods and can only produce a single foreground-to-background binary mask. Ground truth masks was used as foreground seeding for WS, a border box was used as background seeding for GCUT, and the DRLSE mask is the inner region of the level-set function. OTSU mask was generated by directly applying Otsu thresholding on the original image. BING is an objectness measure, we normalise all bounding-box plots on a single image as the saliency map. The latter methods applied to SatSeg are not typically considered as saliency images, we use these as reference measures to gauge the general object segmentation problem.

Evaluation Metrics

We apply ROC [351] as the evaluation metrics for this investigation. For a computed saliency map, S , it is converted to an 8-bit binary mask image, M , by applying a constant threshold. The precision and recall of a single image can be computed from the ground truth by performing the following piece-wise operation to the saliency maps,

$$Prec = \frac{TP}{EP} = \frac{\sum_{i=1}^R \sum_{j=1}^C |M^{(ij)} \cap G^{(ij)}|}{\sum_{i=1}^R \sum_{j=1}^C |M^{(ij)}|}, \quad (4.14)$$

$$Rcal = \frac{TP}{AP} = \frac{\sum_{i=1}^R \sum_{j=1}^C |M^{(ij)} \cap G^{(ij)}|}{\sum_{i=1}^R \sum_{j=1}^C |G^{(ij)}|}, \quad (4.15)$$

where TP , EP , AP are true positive, estimated positive, and actual positive respectively, M_{ij} and G_{ij} are the individual pixels in the saliency and ground truth maps M and G . R and C are the rows and columns of the saliency and ground truth maps. A threshold ranging from 0 to 255 is used to control recall. We compute the standard units of measure for saliency evaluations including average and maximum F -Measure [186] and AUC from the ROC curve. The F -measure is defined as

$$F_{\beta} = \frac{(1 + \beta^2) Prec \times Rcal}{\beta^2 \times Prec + Rcal}, \quad (4.16)$$

where β^2 is set to 0.3 [186]. The ROC curve is generated using the following definitions of True Positive Rate (TPR), where $TPR = Rcal$, and False Positive Rate (FPR), where

$$FPR = \frac{EP - TP}{Total - AP} = \frac{\sum_{i=1}^R \sum_{j=1}^C |M^{(ij)} \cap (255 - G^{(ij)})|}{\sum_{i=1}^R \sum_{j=1}^C |255 - G^{(ij)}|}. \quad (4.17)$$

We use the Mean Absolute Error (MAE) [155] to take into account continuous saliency map variations. Both the saliency map, \bar{S} , and the ground truth map, \bar{G} , are normalised

Table 4.7: Computing Platforms

Platform	Description	Core	OS
32W	Intel 2-Quad-Q6600	4-2.4GHz	WinVista
64W	AMD A4-5000-APU	4-1.5GHz	Win10
64L	AMD FX-8350	8-4.0GHz	Ubuntu16

between the range of $[0, 1]$. The MAE is defined as

$$MAE = \frac{1}{RC} \sum_{i=1}^R \sum_{j=1}^C |\bar{S}^{(ij)} - \bar{G}^{(ij)}|. \quad (4.18)$$

where $\bar{S}^{(ij)}$ and $\bar{G}^{(ij)}$ are the (i, j) elements of $\bar{\mathbf{S}}$ and $\bar{\mathbf{G}}$ respectively.

Computing Platforms

We used two computer platforms in the saliency performance evaluation, a Linux platform and a Windows platform. The descriptions for the computing platforms used in the performance runs are provided in Table 4.7. Two platforms and operating systems were used because some of the comparison models are given in windows compiled executables.

4.2.8 Foreground Extraction Performance

Precision Performance

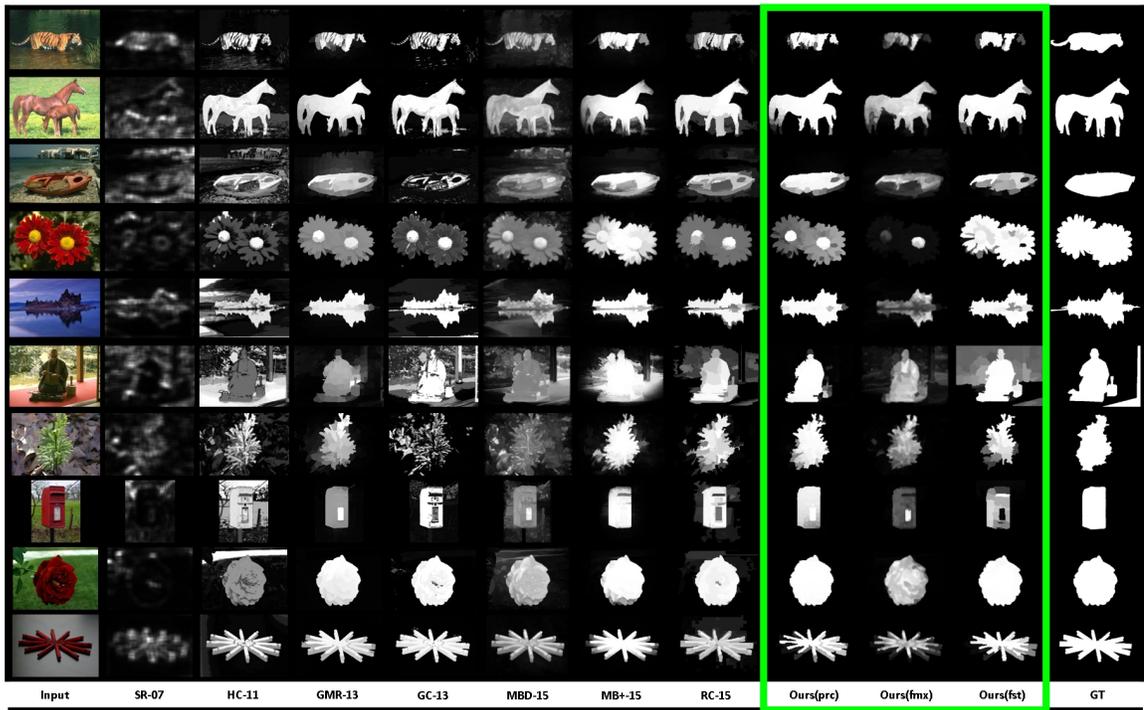
We compared with the benchmark models using three standard datasets ECSSD [298, 299], DUT-OMRON [99], MSRA10K [167, 169] and our SatSeg over a total of 32, 536 test images including colour and grayscale variations. Fig. 4.11 and 4.12 provides selected qualitative results for the standard and SatSeg datasets respectively. Due to space limitations, we only present the highest quality benchmark models for comparison plus the SR model as a reference. Even though our primary focus is to reduce runtime, our models have exceeded the precision performance of most benchmark models in several areas. We point to the *Monk* image in Fig. 4.11(a) and the *Rose* image in Fig. 4.11(b) where our *prc* and *fmx* models out-performs all benchmark techniques. We include Otsu thresholding for the SatSeg comparisons of Fig. 4.12. In the ISS infrared image with only space as the background per

Fig. 4.12 (1st row), the Otsu thresholding resulted in the most accurate foreground extraction. However, when there is a complicated background Earth passage, Otsu thresholding is the most inaccurate method. In the *ISS Blue Earth* (5th row), *ISS Cloud Earth* (6th row), *ISS Solar Panel* (7th row), and *Orbital Express over Earth* (9th row), and *Space Shuttle* (11th row) images, almost all benchmarking models cannot distinguish the Earth background and the foreground vehicle, contrastingly, our *fm*x and *fst* methods almost entirely removed the Earth background. Non-distinctiveness is worsened when only the monochromatic image is available. An example of this is the *Radarsat Model* (10th row); in this case, our *fm*x and *fst* method provide the best match to the ground truth.

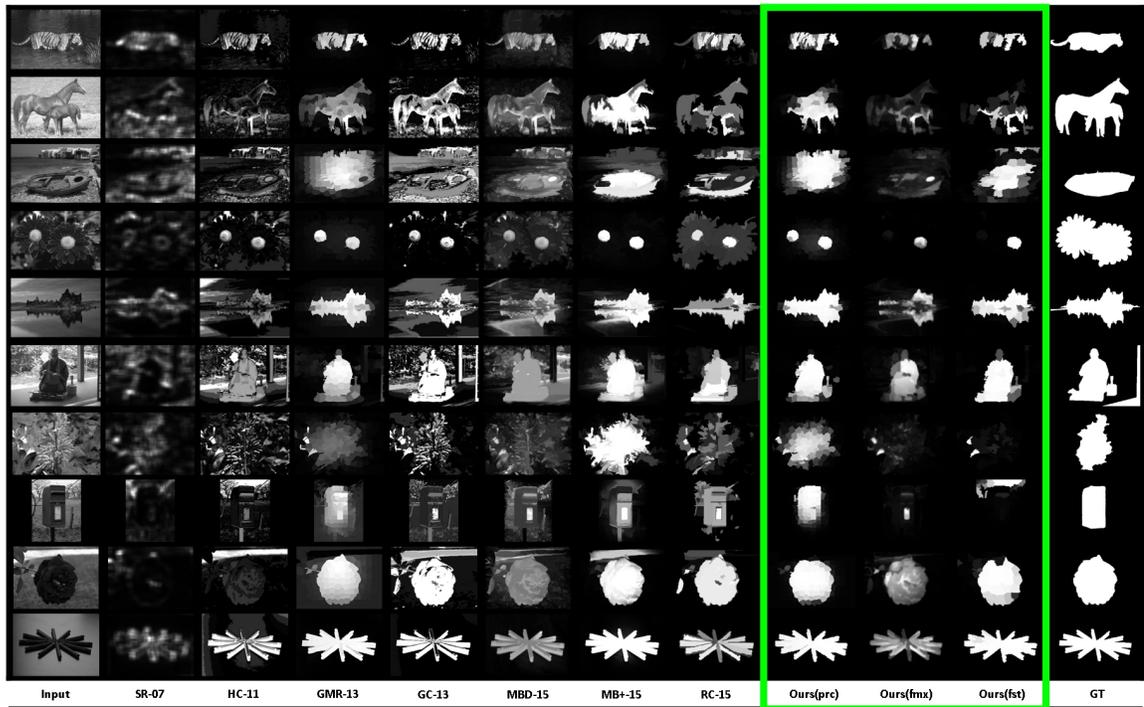
Quantitative results for all datasets are shown in Fig. 4.13 and Fig. 4.14 for colour and grayscale images respectively. For all datasets, benchmark models that rely on colour perform worse when only the grayscale intensity is available. Our *prc* model has the highest mean F -measure and lowest MAE in all standard datasets and is only eclipsed by MB+ in SatSeg for F -measure, MB+ and RC for MAE. Our *fm*x model consistently matches the top performers in all datasets in precision versus recall and AUC. Our *prc* model outperforms all models in the grayscale ECSSD dataset measured by the precision versus recall curve per Fig. 4.14(a). In the grayscale SatSeg dataset in Fig. 4.14, our *fm*x model has a higher precision versus recall performance than all other models. Our *fst* model matches the top RC and MB+ performance and outperforms all other methods in the mean F -measure. OTSU, DRLSE, and BING have much higher MAE than other methods; this is because OTSU and DRLSE are intensity threshold and segmentation methods, while BING is an objectiveness bounding box technique and was not designed to be a salient image generator. The general saliency performance trend increases with increasing publication year. While our models did not achieve a considerable increase in precision performance, we meet or exceed the benchmark precision and significantly exceeded the benchmark model run speeds.

Timing Analysis

A speed test was performed using the average run time for computing the SatSet dataset. The timing requirement is less than 50 ms. Figure 4.15 provides the timing for the various models. Table 4.7 provides the specifications of various computer platforms used in the

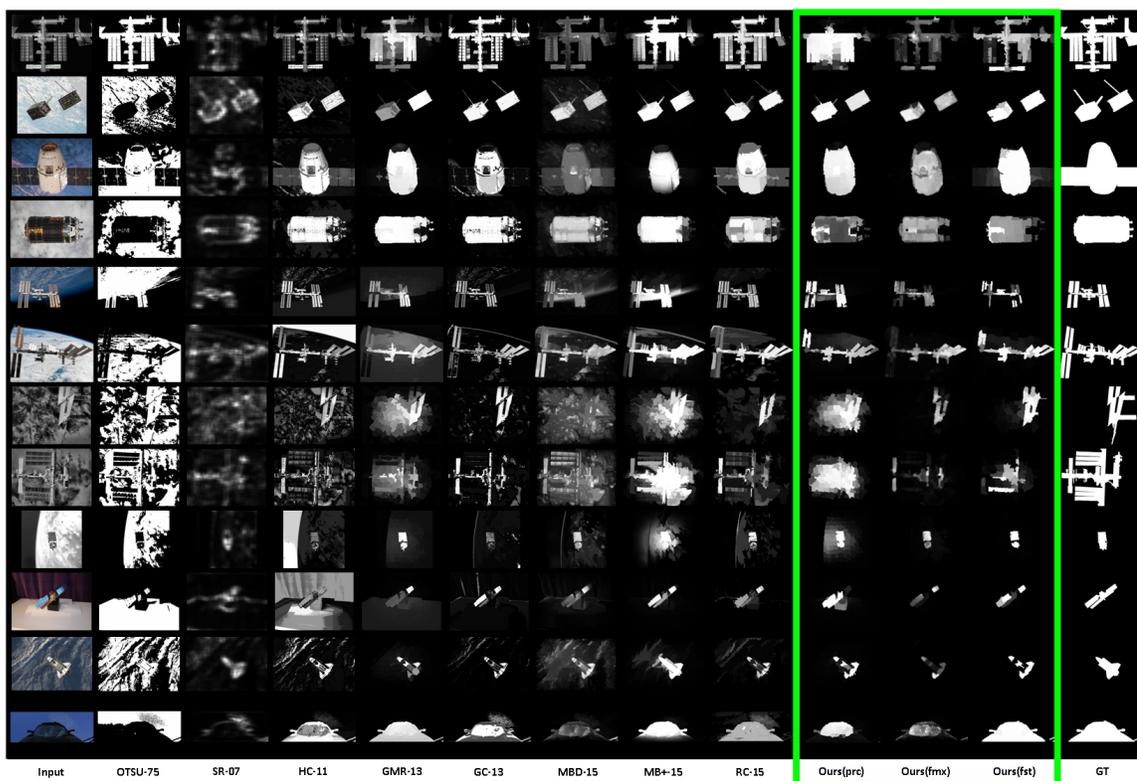


(a) Colour

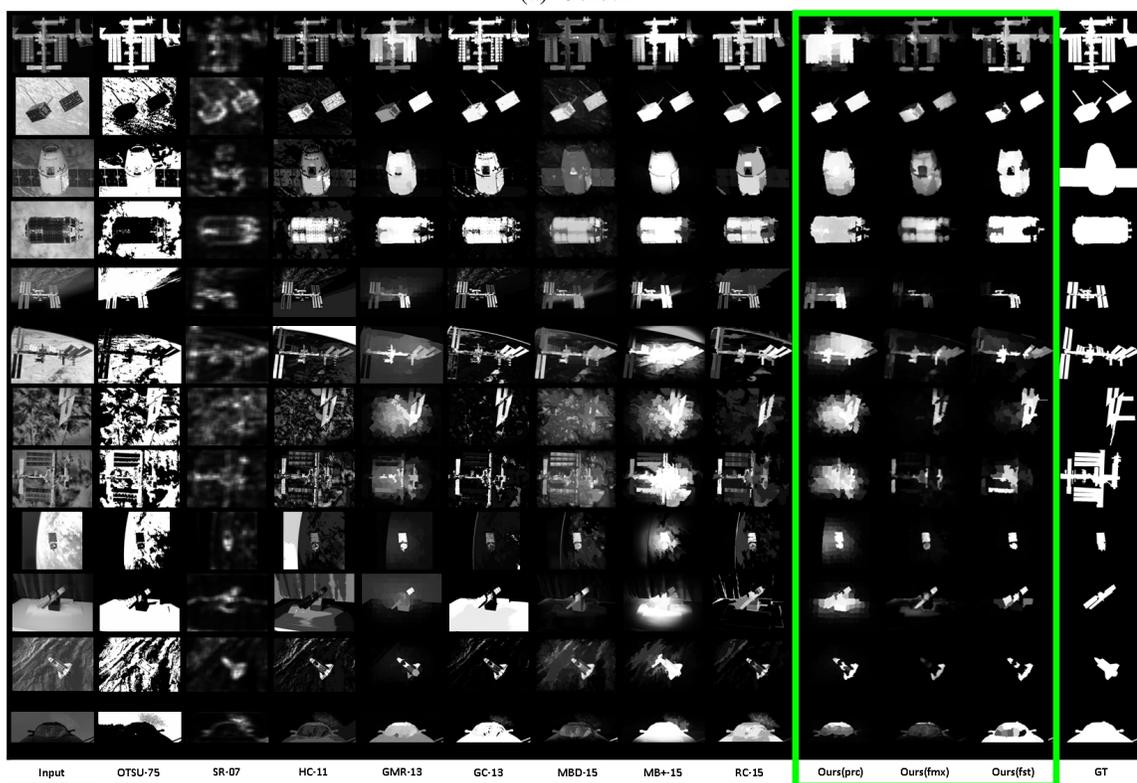


(b) Grayscale

Figure 4.11: Standard dataset salient image comparisons.



(a) Colour



(b) Grayscale

Figure 4.12: SatSeg dataset salient image comparisons.

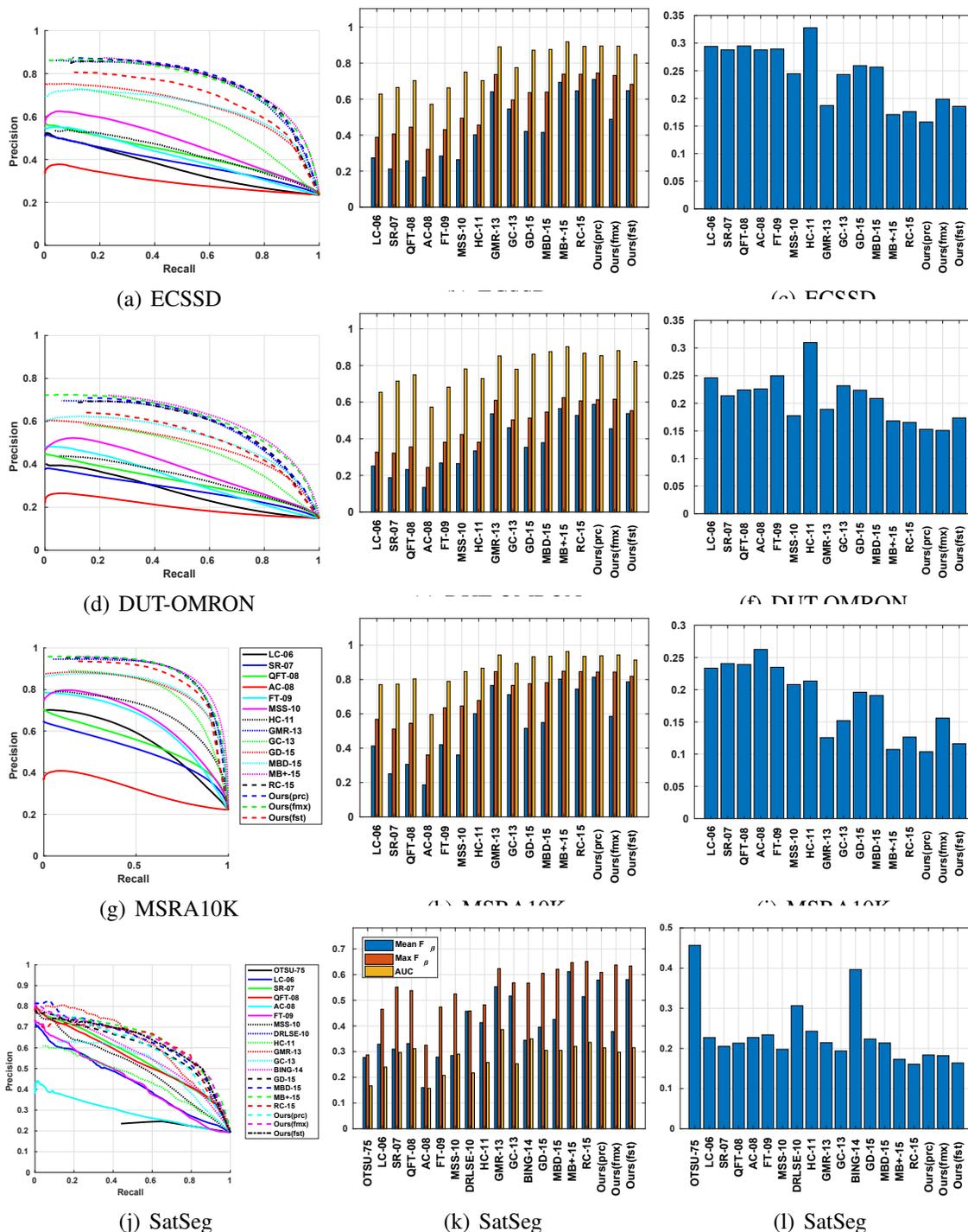


Figure 4.13: Saliency detection results for colour images. Column 1: ECSSD, DUT-OMRON, MSRA10K shares the same legend in 4.13(g). Column 2: all figures shares the same legend in 4.13(k).

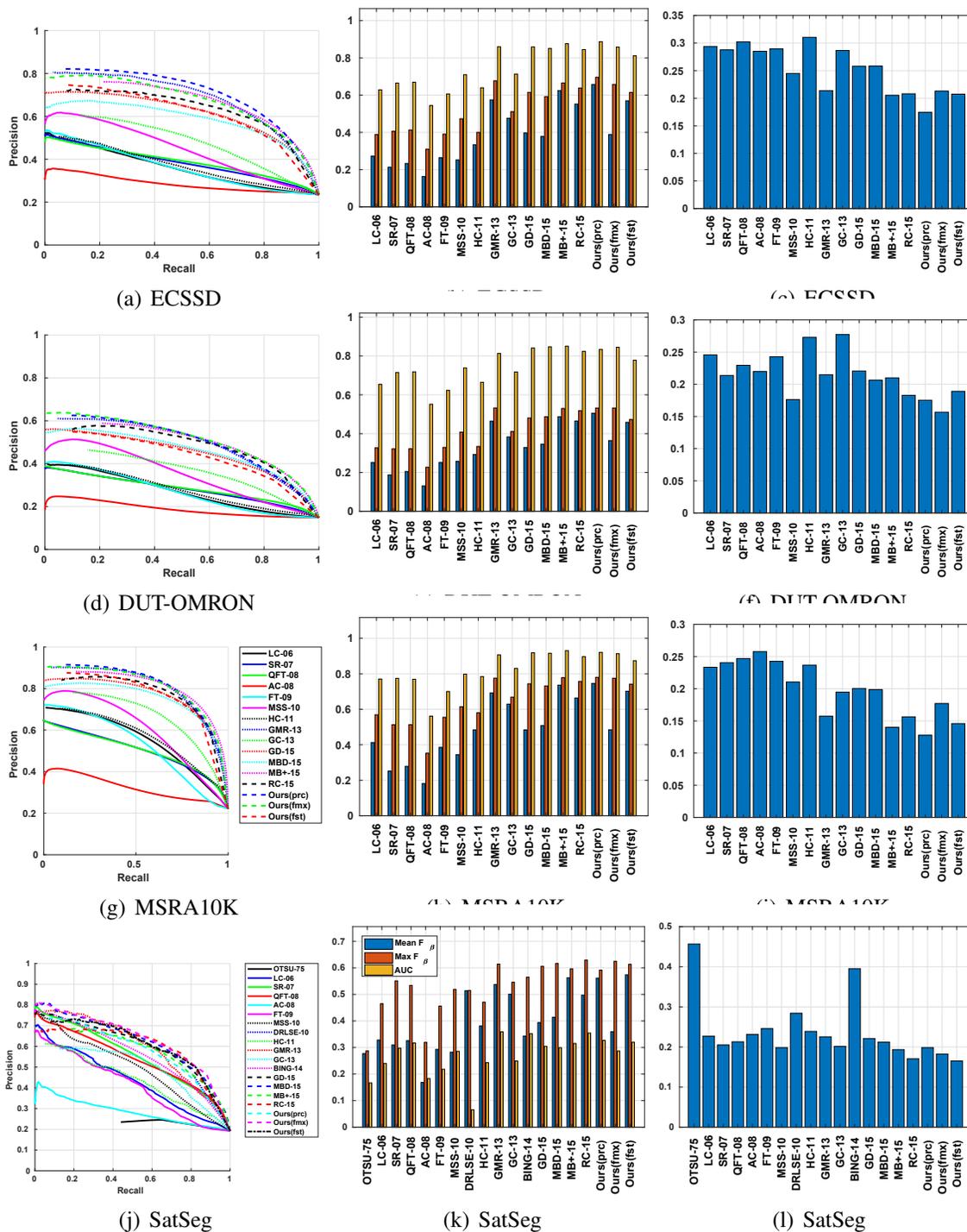


Figure 4.14: Saliency detection results for grayscale images. Column 1: ECSSD, DUT-OMRON, MSRA10K shares the same legend in 4.14(g). Column 2: all figures shares the same legend in 4.14(k).

speed test. Majority of the models are implemented in C++ on the 64L platform. WS, GCUT, OTSU, SR and BING are class modules from OpenCV 3.3. AC, GC, and MB+ are windows compiled executables provided by the various authors that can only be run from the 64W platform. The source code for DRLSE was supplied by Li [285] in MATLAB and was executed on the 32W platform. Figure 4.15 shows the slowest model is DRLSE, it has an average run time of approximately 2.5 minutes per image. While the platform and the coding language skewed the DRLSE timing higher, this model requires time-consuming evolutions of the level-set function and is estimated to be too costly for real-time. MB+ has the fastest timing compared to the other win executables. It is double the requirement of 50 ms running at 98.5 ms per image. It is possible this method could be even faster running on the 64L platform; however, our C++ implementation of the MB+ extensions did not achieve the specified performance by Zhang *et al.* [185]. Both RC and GMR exceeded the 50 ms target with 440 and 500 ms per image respectively. Our *fmx* and *fst* models are more than 10 times faster at 48.122 ± 0.186 and 41.242 ± 0.097 ms respectively. Our *prc* model is still 3 times faster at 145.09 ± 1.197 ms. OTSU, LC, SR, QFT are on the order of 3 to 5 ms but fails to achieve the same precision as our model. Our *fst* and *fmx* models archived the best precision when compared to models with similar speeds.

4.3 Infrared Spacecraft Image Saliency Extraction

From an image processing perspective, there are two main scenarios in the spacecraft rendezvous imagery; these are the nadir pointing and non-nadir pointing phases by the imager. In the latter case, the target spacecraft can easily be extracted using thresholding methods since the background is generally black with some minor polluting light source from stars or camera hardware distortions. Strong lighting source from the Moon or the Sun is also possible, but they are localised and rarely occur. The former scenario is much more difficult to resolve, especially for monochromatic images. The Earth background can clutter the input image with clouds, land patterns, and brightly reflected sunlight from the oceans. Operationally, a nadir view during rendezvous operation is unavoidable [16]. For example, there is only one zenith-wise corridor for logistical vehicles to approach the ISS; therefore the view of the target spacecraft from the ISS will always have the Earth backdrop. Another example is in a geostationary servicing mission; the most logical docking face is on

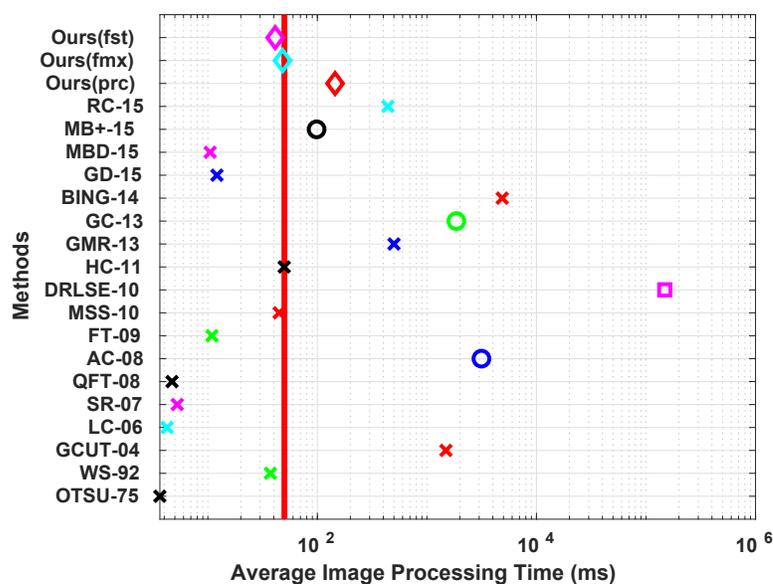


Figure 4.15: Saliency model timing comparisons. Vertical red line indicates the required real-time performance. The \times markers indicate runs computed using the 64L platform with C++; the \circ markers indicate runs calculated using the 64W platform with windows compiled executables, the \square markers indicate runs computed using 32W platform with MATLAB, the \diamond markers indicate our models ran on the 64L platform with C++. Refer to Table 4.6 for method codes. Refer to Table 4.7 for platform codes.

the anti-Earth deck which houses the launch adaptor ring; therefore the servicing vehicle must approach the target satellite from the Nadir direction.

The computation resource required for the two scenarios is very different; the non-nadir view can be computed using fast adaptive thresholding, whereas the nadir view requires more software intelligence to extract the foreground target. We developed a reliable unsupervised foreground extraction technique for the non-nadir scenario. At the same time, this technique detects possible Earth background so the image processing can transfer to a more complex algorithm in separating the foreground vehicle from the Earth.

4.3.1 Background Classification

In the non-nadir pointing scenario, only the backdrop of black space is behind the target vehicle. One may consider the input image already as a form of the foreground saliency map without any additional processing. A closer examination, however, shows there can be various defects in the image. Figure 4.16 shows increasing threshold level of an ISS infrared image. Figure 4.16(a) shows hardware heating regions can be detected in the upper left corner when the threshold level is 10; this is due to thermal sensor bias that is continuously changing over time. Nearly 30 percent of the vehicle is non-visible when the threshold is 100. The threshold level of 60 removes the SSRMS and the lower solar panel. Out of the four images, a threshold of 30 is the closest match to the ground truth (GT) while some of the border sensor noise remains visible.

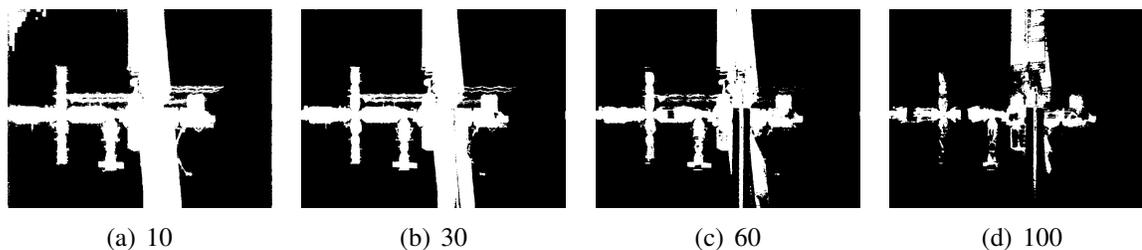


Figure 4.16: ISS infrared image threshold level variation, values under each figure represent the intensity threshold value.

Our method for region detection and background identification is provided in Algorithm 4 and Fig. 4.18 for the procedure and the procedure sequence images respectively. First, we resize the input image, I , to 120 rows and normalise the resized image between

0 to 255, $\bar{\mathbf{I}} = \mathcal{N}(\mathbf{I})$. Next, we apply mean value binary thresholding, $\mathcal{M}(\mathbf{x})$, on the image to remove low-intensity defects such as image sensor local bias and background starlight. If the detector is operating during the nadir phase, the bright Earth background will remain unaffected. A drawback of thresholding is the creation of holes when removing shadowed regions; it will also keep dust particles from image noise. We apply contours on the threshold mask and keep only the maximum perimeter region using the operator $\mathcal{R}(\mathbf{x})$. The main region contour will fill in holes and remove all dust particles from it; we find this to be more effective than using an *open* morphological operation. The main region contour mask, \mathbf{S}_u , has fine border resolution, but it is only the result of illumination intensity, it will not exclude the Earth background if the camera is nadir pointing. To distinguish which direction the camera is pointing to, we use the intensity on either side of the high-frequency response to decide the region class. High-frequency edge features are the result of spacecraft boundaries, internal vehicle connections, Earth horizon, and sharp Earth textures such as coasts and other geological boundaries. During the Earth passage, illuminated regions will occur on both sides of the border feature whereas if the imager is non-nadir pointing, the most brightly lit region will only occur on the inside of the outer spacecraft border. We use this crucial observation as the decision rule to classify the pointing direction. To extract the high frequency content, we first approximate the gradient of the normalised input image, $\mathbf{S}_e = \nabla(\bar{\mathbf{I}})$, by convoluting $\bar{\mathbf{I}}$ with the 3×3 Scharr kernel, \mathbf{K}_s . To extract the local region, we increase the edge responses by taking the DoG. The DoG is a faster numerical approximation of the LoG from the heat equation, and it will diffuse the edge response uniformly in all directions surrounding the high-frequency edges. The normalised response map, \mathbf{S}_d , is a faster, and simplified form of the DoG [15] by convoluting the edge response with the difference of two Gaussian kernels of varying scales. We denote the normalised DoG as $\overline{\text{DOG}}(\mathbf{x})$. We also compare using the linear diffusion heat equation with the non-linear diffusion equation. We compute the non-linear diffusion filtering using FED [10]. We find similar results in precision using both approaches; the DoG approach takes nearly half the FED computation time with only one FED iteration step. Specifically, the DoG runtime for the test image is 1.107 ms and the FED run time is 2.149 ms. Figure 4.17 shows the DoG and FED response maps and the respective final foreground masks. Next, we compute the foreground mask, \mathbf{S}_{fg} , by applying both the Otsu [139] and mean binary threshold to \mathbf{S}_d

and union with S_u to fill large gaps and holes. We can compute the background mask, S_{bg} , by taking the bitwise *NOT* operator of the foreground map. The foreground and background pixel maps, S_f and S_b respectively, are the intersect of \bar{I} and the respective masks. After computing the mean and standard deviation of the foreground and background intensity histograms, we formulate our decision rule for the pointing phase classification as follows,

$$\left(\frac{\mu_b + \sigma_b}{\mu_f + \sigma_f} < r_{tol} \right) \begin{cases} 1 & \text{non-nadir phase (space background)} \\ 0 & \text{nadir phase (Earth passage)} \end{cases}, \quad (4.19)$$

where μ_i and σ_i , $i \in \{f, b\}$, are the mean and standard deviation of the foreground and background intensity histograms respectively. The mean and standard deviation for the background histogram will shift higher if there is Earth passage and vice-versa. Equation (4.19) separates two highly distinctive pointing phase described by the intensity histograms. We select the ratio tolerance to $r_{tol} = 0.2$ in our implementation. If the image class is non-nadir pointing, we confirm the foreground mask by intersecting S_{fg} with the main region from the edge response, S_{fe} . The benefits of the intersection can be observed in Fig. 4.19, where Fig. 4.19(a) is the original image, Figs. 4.19(b) and (c) are the foreground mask intersection without and with S_{fe} respectively. The image computed with S_{fe} has more refined foreground map beneath the right solar panel and around the solar wing connections. If the image class is nadir pointing, more sophisticated saliency detection is needed to extract the foreground spacecraft, refer to Sec. 4.3.2 for details. Finally, we resize the foreground mask to the input image size if required. An automatic thresholding technique by Otsu [139] se-

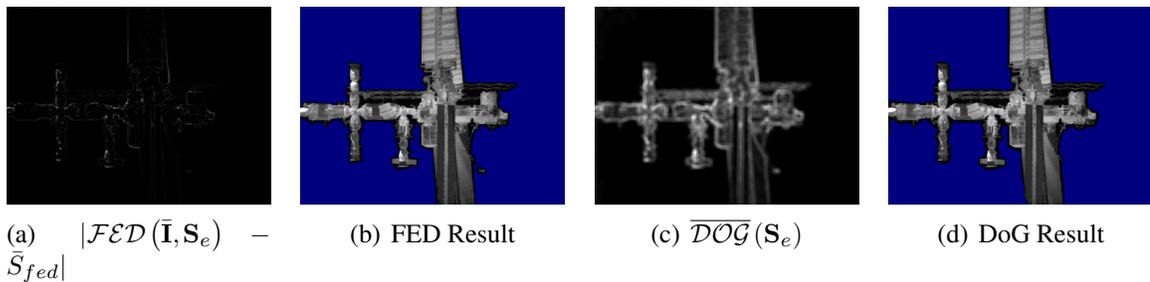


Figure 4.17: FED vs. DoG comparisons. \bar{S}_{fed} is the mean value of $\mathcal{FED}(\bar{I}, S_e)$. The FED results are slightly more accurate by preserving the end-effector on the SSRMS.

lects the optimal separation in the image histogram as the threshold value. Figure 4.20(a) is

Algorithm 4 The *REGION_DETECT* foreground mask algorithm. Algorithm sequence images are provided in Fig. 4.18.

```

1: procedure REGION_DETECT(I)
2:   if Image larger than 120 rows then
3:     Resize image to 120 rows
4:    $\bar{\mathbf{I}} \leftarrow \mathcal{N}(\mathbf{I})$ 
5:    $\mathbf{S}_u \leftarrow \mathcal{R}(\mathcal{M}(\bar{\mathbf{I}}))$ 
6:    $\mathbf{S}_e \leftarrow \mathbf{K}_s^{(3 \times 3)} * \bar{\mathbf{I}}$ 
7:    $\mathbf{S}_d \leftarrow \mathcal{N}\left(\left(\mathbf{G}(\sigma = \sqrt{2}^7) - \mathbf{G}(\sigma = \sqrt{2}^{-7})\right)^{(11 \times 11)} * \mathbf{S}_e\right)$ 
8:    $\mathbf{S}_o \leftarrow \mathcal{R}(\mathcal{O}(\mathbf{S}_d))$ 
9:    $\mathbf{S}_m \leftarrow \mathcal{R}(\mathcal{M}(\mathbf{S}_d))$ 
10:   $\mathbf{S}_{fg} \leftarrow (\mathbf{S}_u \cup \mathbf{S}_o \cup \mathbf{S}_m)$ 
11:   $\mathbf{S}_{bg} \leftarrow \sim(\mathbf{S}_{fg})$ 
12:   $\mathbf{S}_f \leftarrow \bar{\mathbf{I}} \cap \mathbf{S}_{fg}$ 
13:   $\mathbf{S}_b \leftarrow \bar{\mathbf{I}} \cap \mathbf{S}_{bg}$ 
14:  Form intensity histograms from  $\mathbf{S}_f$  and  $\mathbf{S}_b$ .
15:  Compute  $\mu_i$  and  $\sigma_i$ ,  $i \in \{f, b\}$  from histograms.
16:  Compute decision rule per Eq. (4.19).
17:  if non-nadir phase then
18:    Foreground Mask:  $\mathbf{S}_{mask} \leftarrow \mathcal{R}(\mathbf{S}_e) \cap \mathbf{S}_{fg}$ 
19:  else
20:    compute fst+ per Algorithm 6
21:  if Resized then
22:    Resize foreground mask to original size

```

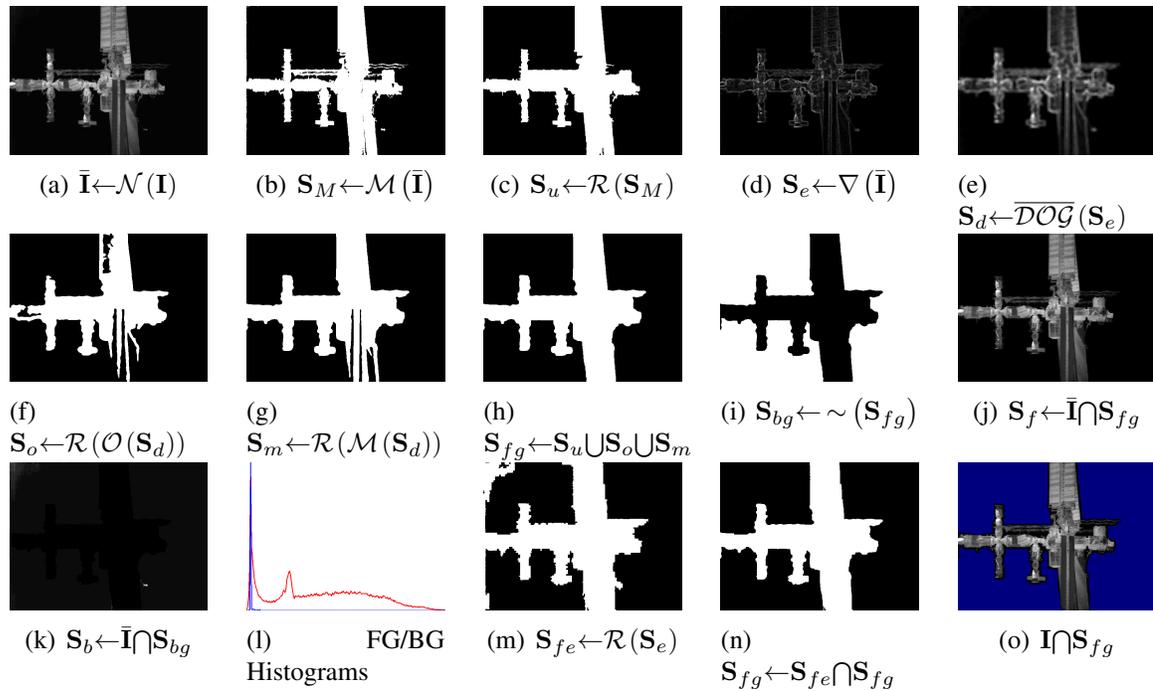


Figure 4.18: Pointing phase identification and foreground mask generation. The foreground and background histograms are red and blue lines in Fig. (l) respectively. Refer to Algorithm 4 for sequence details.

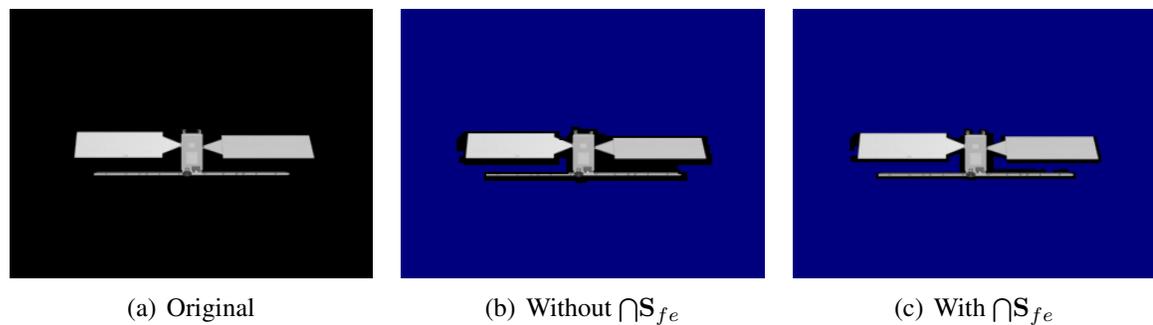


Figure 4.19: Radarsat 3D model test image.

the original image, Fig. 4.20(b) is the Otsu thresholding [139], Fig. 4.20(c) is our region detection method in Algorithm 4 overlaid on top of the original image, the none blue region represents the foreground mask, and Fig. 4.20(d) is the ground truth mask. Our method slightly overpredicts the border region, but do not exclude any spacecraft regions.

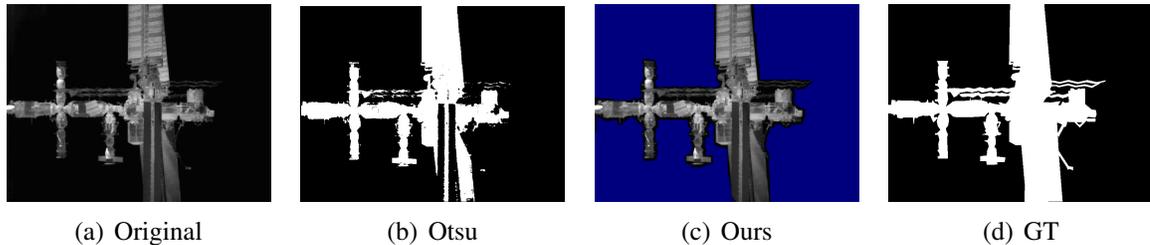


Figure 4.20: ISS infrared image comparison of contrast thresholding and region detection.

4.3.2 Nadir Pointing

The nadir pointing Earth passage view is more difficult to process than the non-nadir pointing images. In addition to a cluttered background, part of the challenge comes from the single intensity channel of the infrared image. In most cases, the foreground pixels are none distinctive from the background and therefore harder to classify. We develop three novel saliency detection procedures for spacecraft foreground extraction from a cluttered Earth background. As previously mentioned, these methods are purely image-driven and do not require data training.

Laplace Operator Response

We develop the *LAPLACE* algorithm as an efficient way to extract the high-frequency response by using the Laplacian and morphological operators. Let us define $CTG(\mathbf{x})$ as a function that converts a colour image into a standard grayscale image, and $\mathcal{B}(\mathbf{x})$ as the box filter operator. We also define $\mathcal{MD}(\mathbf{x})$ and $\mathcal{ME}(\mathbf{x})$ as the morphological dilation and erosion operators using an elliptical kernel respectively. The box filter and morphological operators are used to limit image noise and enlarge the high-frequency response regions. The *LAPLACE* algorithm is given in Algorithm 5.

Algorithm 5 The *LAPLACE* foreground mask algorithm.

```

1: procedure LAPLACE(I)
2:    $\mathbf{S}_L \leftarrow \mathcal{O}(\mathcal{N}(|\mathcal{B}(\mathcal{L}(\mathcal{CTG}(\mathbf{I}), K_L = 3), K_B = 5)|))$ 
3:   for  $i = 1 : (K = 3)$  do
4:      $\mathbf{S}_L \leftarrow \mathcal{MD}(\mathbf{S}_L, K_D = 10)$ 
5:   for  $i = 1 : (K = 2)$  do
6:      $\mathbf{S}_L \leftarrow \mathcal{ME}(\mathbf{S}_L, K_D = 10)$ 
7:   Foreground Mask:  $\mathbf{S}_L \leftarrow \mathcal{R}(\mathbf{S}_L)$ 

```

The *fst+* Algorithm

The *High-frequency Saliency feature* (HiSafe), \mathbf{S}_{HSF} , is computed by normalising the combined SR response and the Laplace filtered response. We also leverage saliency map generated using a modified colour Regional Contrast (RC) [167], $\bar{\mathcal{RC}}(\mathbf{x})$, and Minimum Boundary Distance (MBD) [185], $\mathcal{MBD}(\mathbf{x})$. For a faster implementation of RC, we replace the region segmentation with the SLIC superpixels. The best estimate of the foreground map is the Otsu [139] threshold of the Laplace response map, \mathbf{S}_L , unioned with the mean value binary threshold on the piece-wise multiplication of the Gaussian, HiSafe, MBD, and RC response maps as shown in step 13 of Algorithm 2. We compute both foreground and background GMR response and combine into one final saliency map, \mathbf{S}_{fg} , and apply a sigmoid function to enhance contrast [185]. The foreground mask can be computed by applying the mean binary threshold operation on \mathbf{S}_{fg} .

Algorithm 2 has demonstrated a net increase in performance and significant speed improvements from the original GMR when evaluated on colour and grayscale images. It, however, still lacks the desired precision when applied to the infrared images from the STS-135 ISS flight mission. An extended version of the *fst* algorithm was developed to improve *fst* performance for infrared sensor images. We designate this extended version as *fst+* and provide the procedure in Algorithm 6. The main addition to the extended *fst* model is by reusing the edge response maps computed from Algorithm 4. The DoG threshold responses add more confidence to the *fst* foreground prediction and do not add more computation time. Finally, the best estimate foreground response is added to the GMR saliency response to adjust the coarse resolution output from the SLIC superpixels.

Algorithm 6 The *fst+* saliency map and foreground mask algorithm. Gray coloured text are the original *fst* algorithm, black coloured text are the extended addition to the *fst* algorithm.

```

1: procedure FST_PLUS(I)
2:   Check for grayscale image
3:    $\mathbf{I}_{Lab} \leftarrow \mathcal{LAB}(\mathbf{I})$ 
4:    $\mathbf{S}_S \leftarrow \mathcal{SLIC}(\mathbf{I}_{Lab}, N = 97)$ 
5:    $\bar{\mathbf{E}} \leftarrow \mathcal{E}(\mathbf{S}_S)$ 
6:    $\mathbf{S}_{SR} \leftarrow \mathcal{N}(\mathcal{SR}(\mathbf{I}, \sigma = 2))$ 
7:    $\mathbf{S}_G \leftarrow \mathcal{G}(\mathcal{MC}(\mathbf{S}_{SR}))$ 
8:    $\mathbf{S}_L \leftarrow \mathcal{N}(|\mathcal{B}(|\mathcal{L}(\mathbf{I}, K_L = 3)|, K_B = 5)|)$ 
9:    $\mathbf{S}_{HSF} \leftarrow \mathcal{N}(\mathbf{S}_{SR} + \mathbf{S}_L)$ 
10:   $\mathbf{S}_{HSF} \leftarrow \mathbf{S}_{HSF} + \text{Mean}(\mathbf{S}_{HSF})$ 
11:   $\mathbf{S}_{MBD} \leftarrow \mathcal{N}(\mathcal{MBD}(\mathbf{I}_{Lab}))$ 
12:   $\mathbf{S}_{RC} \leftarrow \overline{\mathcal{RC}}(\mathbf{I}_{Lab})$ 
13:   $\mathbf{S}_{fg} \leftarrow \mathcal{O}(\mathbf{S}_L)/255 + \mathcal{M}(\mathbf{S}_G \circ \mathbf{S}_{HSF} \circ \mathbf{S}_{MBD} \circ \mathbf{S}_{RC})/255$ 
14:   $\mathbf{S}_{fg} \leftarrow \mathbf{S}_{fg} + \mathbf{S}_o/255 + \mathbf{S}_m/255$ , where  $\mathbf{S}_o$  and  $\mathbf{S}_m$  are from Algorithm 4.
15:   $\mathbf{S}_p \leftarrow (\mathbf{S}_{fg} > K)$ , where  $K = 1$ .
16:   $\mathbf{S}_{fg} \leftarrow \mathcal{N}(\overline{\mathcal{GMR}}(\mathbf{S}_{fg}) \circ (255 - \overline{\mathcal{GMR}}(255 - \mathbf{S}_{fg})))$ 
17:  for  $i = 1 : \mathbf{S}_{fg}$  height do
18:    for  $j = 1 : \mathbf{S}_{fg}$  width do
19:       $\mathbf{S}_{fg}(i, j) \leftarrow \frac{1}{1 + e^{-b(\mathbf{S}_{fg}(i, j) - 0.5)}}$  where  $b = 10$ 
20:   $\mathbf{S}_{fg} \leftarrow \mathcal{N}(\mathbf{S}_{fg})$ 
21:  Foreground Saliency Map:  $\mathbf{S}_{fg} \leftarrow \mathbf{S}_{fg} + \mathbf{S}_p$ 
22:  Foreground Mask:  $\mathbf{S}_{mask} \leftarrow \mathcal{M}(\mathbf{S}_{fg})$ 

```

4.3.3 False-colour High-Frequency Saliency Feature Image

Some infrared images are challenging to separate its foreground from its background. An example is the RSM video, where RSM is spinning about the pitch axis against a curtain backdrop in a laboratory setting. Figure 4.21(a) is the original monocular infrared image captured under laboratory heating conditions. Figure 4.21(b) is the ground truth mask for the input image. Figure 4.21(c) is the PWP3D [101] probability posterior computed using the *prior* mask histogram template. A 3D projection mask based on the initial ground truth pose is used as the *prior* mask. Figures 4.21(d), (e), (f) are the saliency map computed using GMR [99], MB+ [185], and RC [167] methods respectively. Figures (g) and (h), are the saliency map and foreground mask computed using the *fst* [300] and *fst+* [300] respectively. Figures 4.21(i) to (k) are the FC-HSF red, green and blue channels for Scharr

edge gradient, extended border, and region surround maps respectively. Figure 4.21(l) is the final combined FC-HSF image. Figure 4.21(m) is the normalised foreground and background FC-HSF image histogram. Figure 4.21(n) is the PWP3D [101] probability posterior computed using the FC-HSF image. Figure 4.21(o) is the estimated 3D model mesh projection on the original input image. Figure 4.21(p) is the posterior mask of the 3D projection. The various saliency methods and developed models failed to generate a precise RSM foreground response as shown in Figs. 4.21(c) to (h) for level-set probability posterior [300], GMR [99], MB+ [185], RC [167], *fst* [300], and *fst+* [300] methods respectively. Furthermore, since the foreground and background image intensities are near each other, different pose may have similar histogram profile as the initial template. An example is shown in Fig. 4.25, where the first column starting from row 2 shows the level-set pose estimation [300] gradient descent as 3D mesh projections on the input image for various iteration steps. The second column in Fig. 4.25 shows the foreground and background image histogram profiles for the respective iteration step. After four iterations the gradient descent converged to an incorrect local minimum pose, the histogram profile of the converged pose is similar to the initial template histogram profile in Fig. 4.25 column 2 row 1.

To increase distinctiveness in the histogram profile and image, we add the gradient and gradient region to the input image as red and blue colour channels. We also dilate the prior mask to create a probable RoI to clear away the clutters in the background. We cannot use the prior mask directly, because it will cause the pose estimation to converge to the prior foreground instead of the actual input. Since this method uses all three colour channels to generate a false coloured image, and it uses the high-frequency content of the image, we call this approach the False-Coloured High-frequency Saliency Feature (FC-HSF) image. The procedures for creating the FC-HSF is provided as follows: let us define \mathbf{S}_{K-1} as the prior mask, and $\mathcal{CL}(\mathbf{x}, C_L)$ as the CLAHE [317] function, where C_L is the clip limit. The CLAHE operation balances the image intensity in the entire local regions and enhances contrast without being influenced by extreme global intensities. We perform the box filter operator on the CLAHE filtered image to limit noise. We then compute the gradient approximation by convoluting with the 3×3 Scharr kernel, \mathbf{K}_s . Note the gradient approximation, $\tilde{\nabla}(\mathbf{x})$, is not threshold binarised. The strength of the gradient is an useful detail in

characterising the target object and the background scene in the red channel histogram. We extract intensities greater than the mean value from the gradient response denoted by $\tilde{\mathcal{M}}(\mathbf{x})$ to avoid weak edges from the background. We also compute the image region by taking the Otsu threshold [139] of the gradient image; this results in a binarised edge response map. From testing, we find the Otsu threshold [139] gives a more conservative response than using the mean threshold. Let us define $\mathcal{C}(\mathbf{x})$ as an operator for finding and drawing closed regions in an image as a filled area. Practically, this can be achieved using the *findContours* and *drawContours* functions in OpenCV 3.3*. Applying $\mathcal{C}(\mathbf{x})$ on the Otsu threshold [139] edge image produces target object inner surround regions with precise boundary definitions. We intersect both the red and blue channels with the dilated prior, \mathbf{S}_D , to remove image clutter. Next, we intersect the CLAHE filtered input image with the dilated prior. If we only use the inner region based on the dilated prior, a sharp contrast at the region boundary from all the image channels will cause a strong barrier in the convergence process and cause too much prediction reliance on it. To avoid the over-reliance, we smooth the border by outwardly extending the same pixel value at the border to all the zero regions in the green channel image. The red and blue channels are not affected because they do not have a range of pixel intensities at the border of the dilated prior. Let us define $\mathbf{BE}(\mathbf{x})$ as an operator that extends the inner border pixels to the image boundary; this is achieved by applying the *copyMakeBorder* function with the *BORDER_REPLICATE* option in OpenCV 3.3. We perform histogram equalisation (HE), $\mathcal{HE}(\mathbf{x})$, on the border extended image to gain additional contrast enhancement. The histogram equalised response map has stronger contrast and is faster than the CLAHE response. Finally, we combine all the colour channels into a false coloured image using the $\mathcal{GTC}(\mathbf{R}, \mathbf{G}, \mathbf{B})$ operator. This false-coloured image enhances the high-frequency edge response [385] and centre-surround [84] in the input grayscale image. The FC-HSF procedure is summarised in Algorithm 7.

Figures 4.21 (i) to (k) shows the red, green, and blue colour channel images for the RSM respectively. Adjustable weighting factors, $\{C_r, C_g, C_b\}$, may be applied to each of the respective colour channels independently. In our example, all the adjustable weighting factors are set to one. The combined HC-HSF and its histogram template are shown in Figs. 4.21(l) and (m) respectively. Figure 4.21(n) shows a probability posterior foreground

*<https://opencv.org>

Algorithm 7 The *FC_HSF* Algorithm to generate FC-HSF Image.

```

1: procedure FC_HSF( $\mathbf{I}, \mathbf{S}_{K-1}$ )
2:    $\mathbf{S}_D \leftarrow \mathcal{MD}(\mathbf{S}_{K-1}, K_D = 36)$ 
3:    $\mathbf{S}_g \leftarrow \mathcal{CTG}(\mathbf{I})$ 
4:    $\mathbf{S}_e \leftarrow \tilde{\nabla}(\mathcal{B}(\mathcal{CL}(\mathbf{S}_g, C_L = 4), K_B = 3))$ 
5:    $\mathbf{S}_r \leftarrow \tilde{\mathcal{M}}(\mathbf{S}_e) \cap \mathbf{S}_D$ 
6:    $\mathbf{S}_b \leftarrow \mathcal{C}(\mathcal{O}(\mathbf{S}_e) \cap \mathbf{S}_D)$ 
7:    $\mathbf{S}_g \leftarrow \mathcal{HE}(\mathcal{BE}(\mathbf{S}_g))$ 
8:   FC-HSF Image:  $\mathbf{S}_I \leftarrow \mathcal{GTC}(\mathbf{S}_r, \mathbf{S}_g, \mathbf{S}_b)$ 

```

mask computed using the new histogram template. The new foreground mask is more precise when comparing Fig. 4.21(n) with Fig. 4.21(c). The improvement is mainly due to the blue channel region map enhancing foreground and background contrast. We can further improve the posterior mask by using the pose projection from Fig. 4.21(o) as the final mask shown in Fig. 4.21(p).

4.3.4 Algorithm Performance

Saliency Performance

we use laboratory video of a RadarSat Model (RSM) spinning about its pitch axis captured by a 320×240 ICI-9320P infrared camera. Flight images are from the docking and undocking phase of STS-135 mission captured by a 640×480 infrared camera in the *Neptec TriDAR* unit.

Figure 4.22 provides qualitative comparisons of SatSeg saliency images generated by our method boxed in green, compared with several traditional and state-of-the-art saliency models. Figure 4.22 from left to right columns are: the Otsu thresholding (OTSU-75) [139], Spectral Residual (SR-07) [180], Graph Manifold Ranking (GMR-13) [99], Minimum Barrier Distance (MB+-15) [185], and Regional Contrast (RC-15) [167], our *fst+* Algorithm 6, and the Ground Truth (GT). The Otsu thresholding [139] is not typically considered as a saliency detection model; it is included in the comparison to demonstrate while it can generate highly precise foreground map in the non-nadir pointing phase, it has the worst error in the nadir pointing phase when there is a cluttered background. The SR [180] model provide useful attention location cues but does not produce sufficient detail of the target

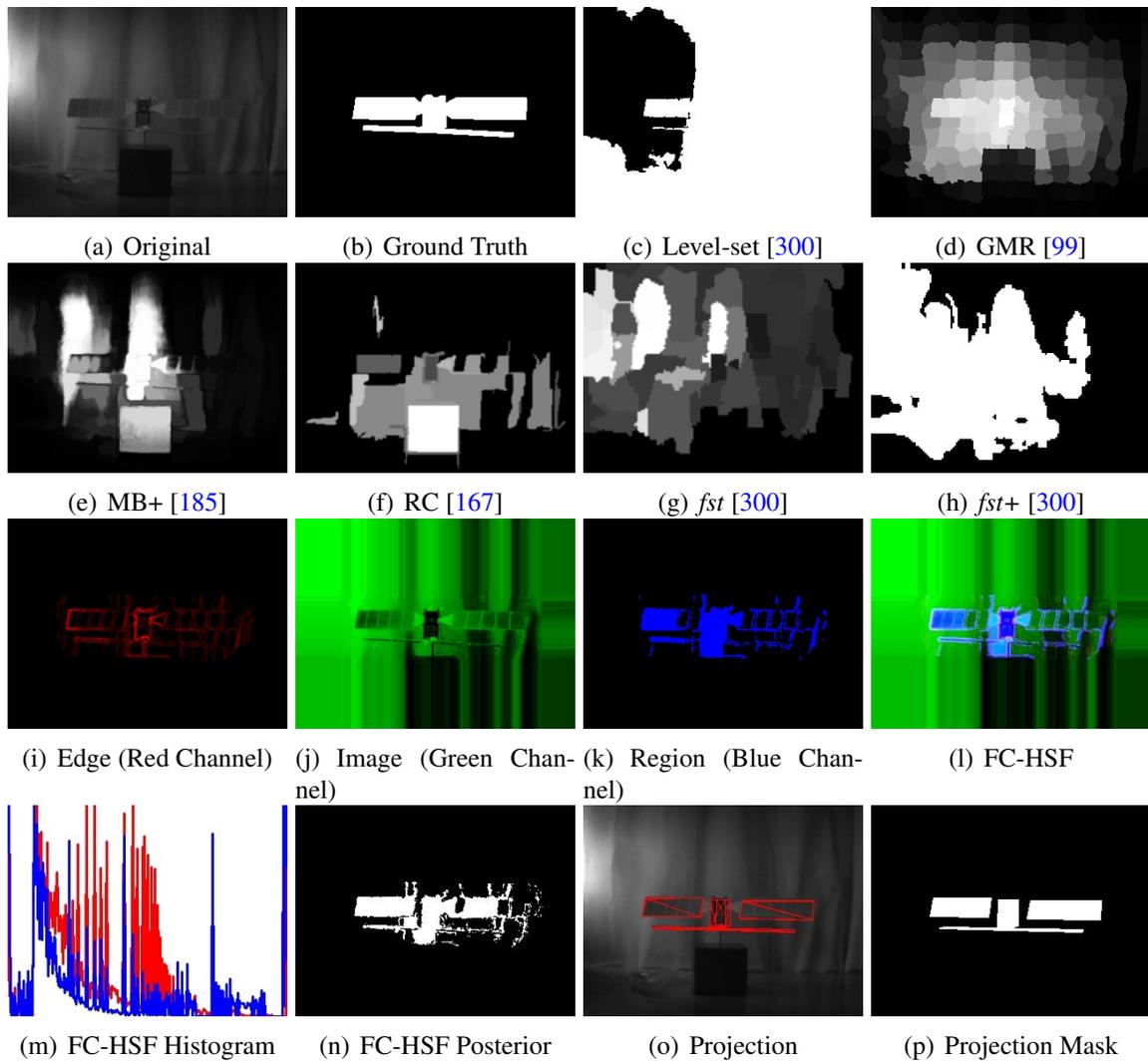


Figure 4.21: RSM infrared image pre-processing and level-set based pose estimation.

object. GMR [99], RC [167], and MB+ [185] are state-of-the-art saliency methods that have good performance. However, both RC and GMR run-times are in the half-second range, and MB+ requires additional segmentation to precisely extract the foreground mask. Figure 4.22 shows our method having the least background error while maintaining the acceptable resolution of the foreground region.

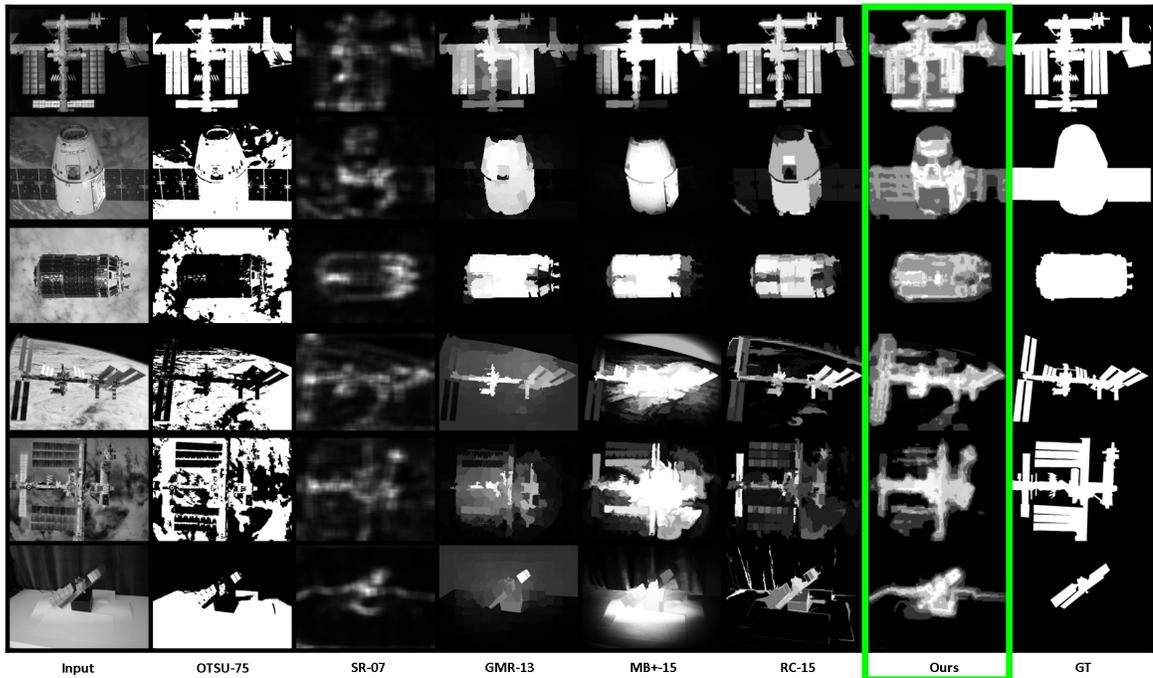
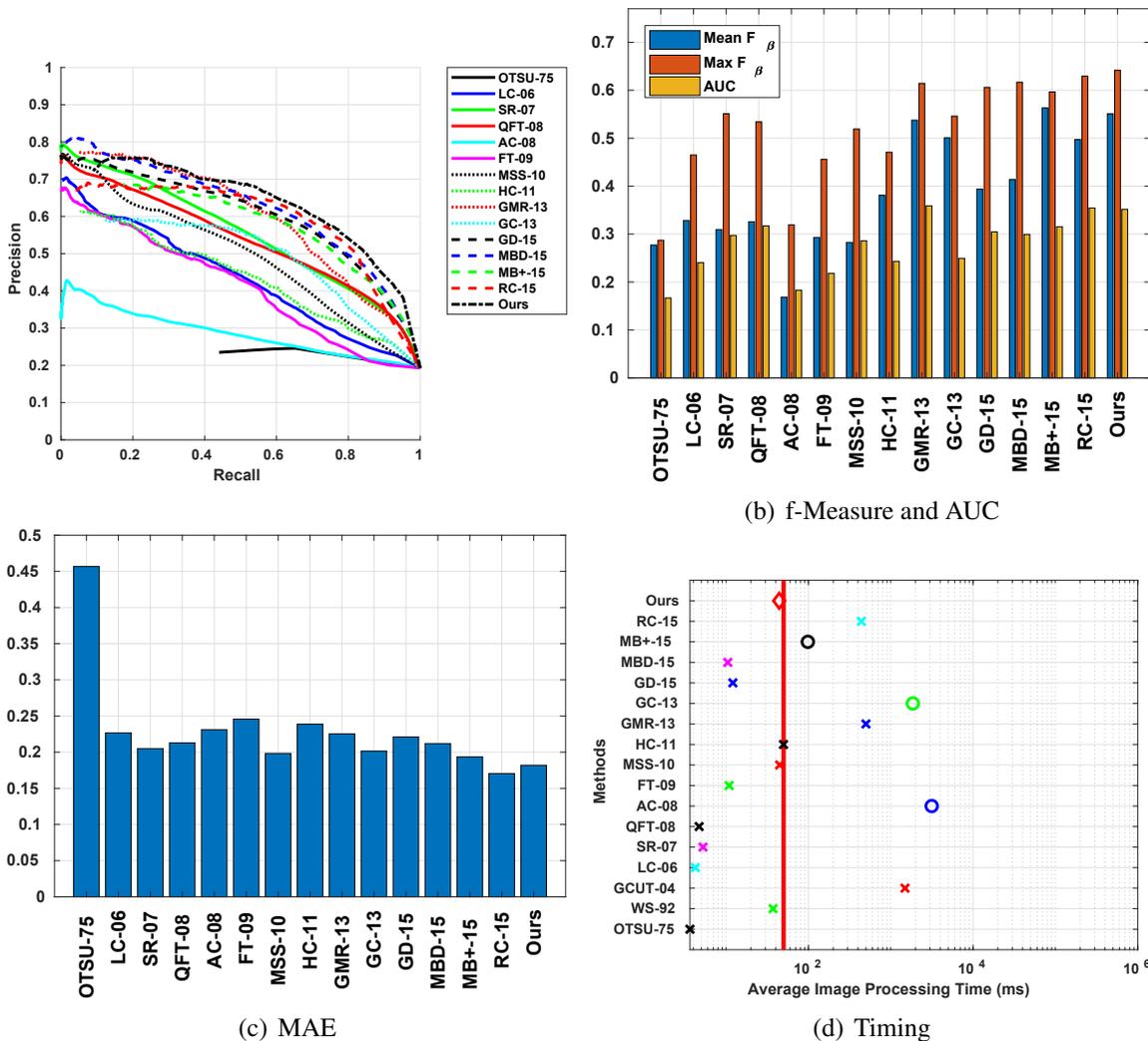


Figure 4.22: Saliency map comparison of selected images from the grayscale SatSeg dataset. The ‘Ours’ label represent the *fst+* method.

Quantitative performance plots are provided in Fig. 4.23. Figures 4.23(a) and (b) provides precision versus recall, mean F -measure, max F -measure, and AUC for 14 traditional and state-of-the-art saliency detection methods plus our *fst+* method respectively. Figures 4.23(c) and (d) provides MAE and timing analysis for single image average run time of the SatSeg dataset; \times and \diamond marker runs are executed using C++/Linux platform; \circ marker runs are executed using built executables on the Windows platform; the red line represents the design timing requirement for the saliency algorithm. Our *fst+* model shows the highest precision versus recall and maximum F -measure performance compared to all the methods. Our mean F -measure and MAE are second to RC and our AUC is comparable to RC and GMR. Speedwise, our *fst+* model average computation time for the SatSeg

dataset is 44.223 ± 0.309 ms; this is $11\times$ faster than the original GMR, $10\times$ faster than RC, and $2.2\times$ faster than MB+. Faster methods such as MBD, GD, FT, QFT, SR, and LC has much lower precision than our *fst+* method. WS, GCUT and OTSU are not saliency detection methods. OTSU thresholding has the worst performance since it does not make any distinction in identifying the foreground and background. GCUT is a Graph Cut [375] segmentation method that uses the image border as the background seed. GCUT's computation time is $33\times$ higher than our method and cannot be used for real-time applications. Watershed (WS) [383] with border seeding do not provide useable foreground maps, we include WS timing for completeness. In summary, Figs. 4.22 and 4.23 shows our *fst+* model to have the best performance overall.

We apply our saliency algorithms to the *TriDAR* infrared camera video from the STS-135 mission ISS undock and fly-by phase. Figure 4.24 shows the input video and the results of the various methods; from left to right, the first column is the original ISS infrared video; the second column is the foreground extraction using Algorithm 4 on the entire video; the third column is using Algorithm 4 during the non-nadir phase and Algorithm 2 during the nadir phase; the fourth column is using Algorithm 5 for the entire video; the fifth column is using Algorithm 4 during the non-nadir phase and Algorithm 6 during the nadir phase. The *REGION_DETECT* method over predicts while the *fst* method under predicts the foreground region. The *LAPLACE* method performed better than the more complex and relatively more expensive *REGION_DETECT* and *fst*. However, the *LAPLACE* method can develop isolated holes or blocks as a result of the morphological operator kernel. The *fst+* method has the best foreground extraction performance overall; it can handle the majority of the infrared video by accurately extracting only the ISS image. The only exception is in a section of the Earth passage (row 3 to 4 in Fig. 4.24) where the background developed sharp edges from the cloud regions; this section of the video is problematic for all the methods, where the ISS is mostly outside the viewable frame with only the solar panel being partially visible. The *fst+* method has the best prediction of the solar panels but over predicted the foreground region by including the cloud regions; this failure is due to a purely image-driven saliency detection process is incapable of classifying useful and unuseful high-frequency contents. Future work may include temporal data with some top-down guidance to increase precision.



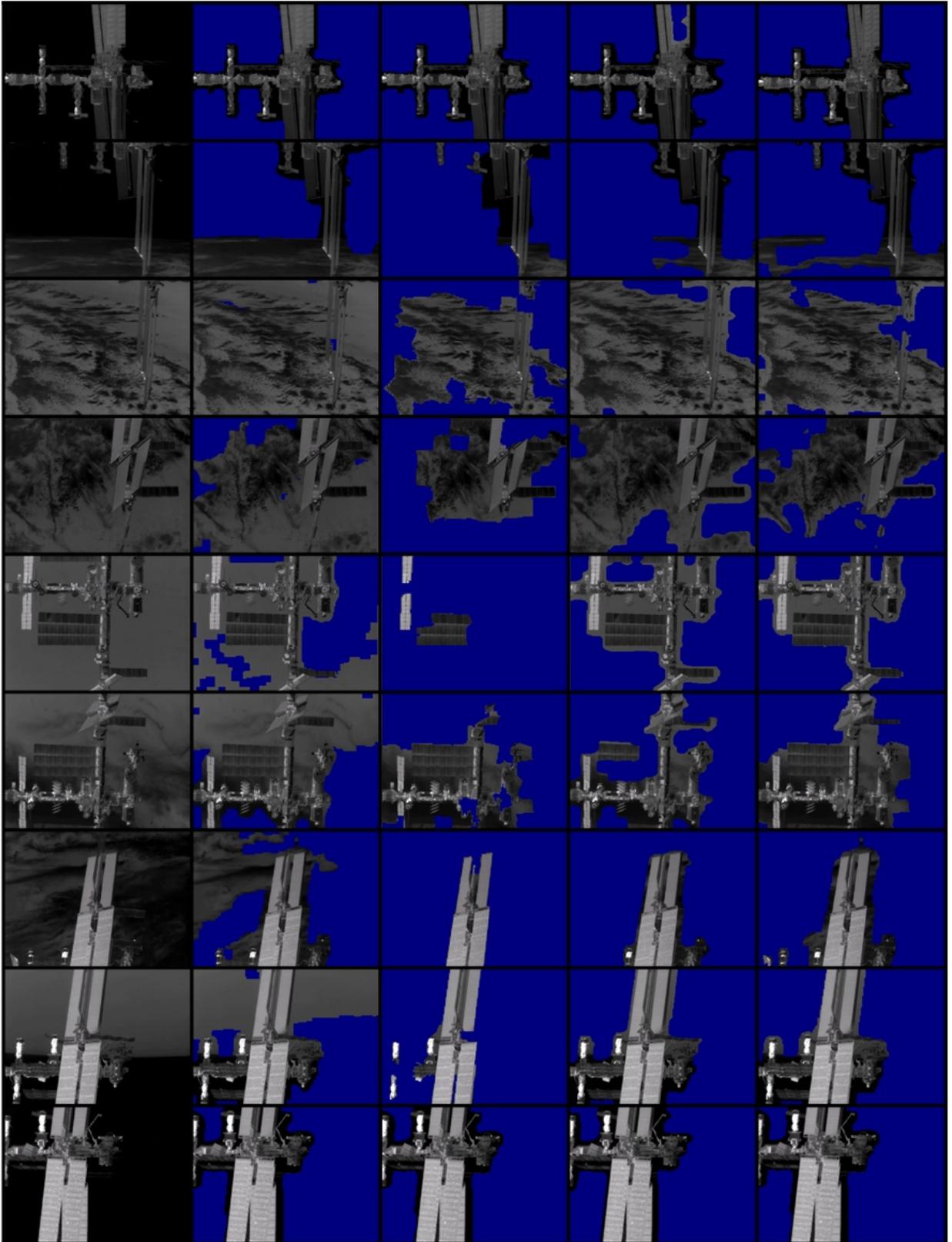


Figure 4.24: ISS saliency method comparisons. For the ISS infrared video test, Algorithm 6 provides the best foreground extraction out of all techniques.

False Colour High-frequency Saliency Features

We apply FC-HSF to the initialisation example shown in Fig. 4.25. Figure 4.25 column 1 and 2 represent single image pose estimation on the original input image without any image processing. Figure 4.25 column 3 and 4 are the pose estimation on FC-HSF images. Figure 4.25 column 1 shows the 3D projection mesh overlay on the original image over the entire gradient descent process. Figure 4.25 column 2 shows the normalised foreground and background histogram of the estimated pose in red and blue lines respectively. Figure 4.25 column 3 shows the projection silhouette overlay on the FC-HSF image. Figure 4.25 column 4 shows the normalised foreground and background histogram of the FC-HSF image in red and blue lines respectively. Figure 4.25 row 1 image 1 shows the prior projection on the input image. Figure 4.25 row 1 image 2 shows the template histogram foreground and background generated using the prior mask on the unprocessed initial input in red and blue lines respectively. Figure 4.25 row 1 image 3 shows the foreground FC-HSF template image. Figure 4.25 row 1 image 4 shows the foreground and background template histogram on the FC-HSF image using red and blue lines respectively. Figure 4.25 row 2 to 8 are gradient descent steps 0, 2, 4, 6, 10, 20, and 30 respectively. In Fig. 4.25, the initial pose misalignment is 0.05 meters in each of the X, Y, and Z spacecraft body coordinates, and 0.05 degrees in each of the Pitch-Yaw-Roll Euler rotation sequence from \mathcal{F}_b . Figure 4.25 show a more distinctive histogram profile, and the FC-HSF solution converges to the correct pose, and the unaltered image does not.

Figure 4.26 shows a 360 degree rotation sequence for the RSM infrared image[†]. The top four rows in Fig. 4.26 represent rotations from 0 degree to 179 degrees; the bottom four rows represent rotations from 180 degrees to 359 degrees. All images in Fig. 4.26 are selected based on equal angle separations. Figure 4.26 row 1 is the 3D CAD projection overlay on the original image. Figure 4.26 row 2 is the FC-HSF image. Figure 4.26 row 3 is the normalised estimated pose foreground and background histogram of the FC-HSF image in red and blue lines respectively. Figure 4.26 row 4 is the level-set function distance map transformation for each of the estimated pose. The red edge lines are the 3D model image projection using the estimated pose. We use the FC-HSF image and histogram profile for

[†]Dataset and video are available at <http://ai-automata.ca/research/hisafe.html>

PWP3D level-set-based pose estimation gradient descent [300]. The approximated level-set function, Φ , is shown as a distance map below the histogram plots. The *zeroth* level-set contour is plotted in a white line while distance inside and outside the zero-level are plotted in red and blue shades respectively. Brighter shade indicates a closer distance to the spacecraft boundary contour. The estimated pose tracked well against the actual model while the highest observed errors occurred during 90 degrees and 270 degrees. Over these two rotation ranges, the visible area is the smallest in the entire sequence. We compared using unmodified image input and FC-HSF image input and found the unmodified image input method diverge from the actual pose shortly after the first image. We also compared using the initial image as the histogram template and using sequential posteriors as the template. The results show the former performed better due to accumulated errors from sequential frames that quickly diluted the following template histograms, suggesting the level-set method is sensitive to the prior mask accuracy.

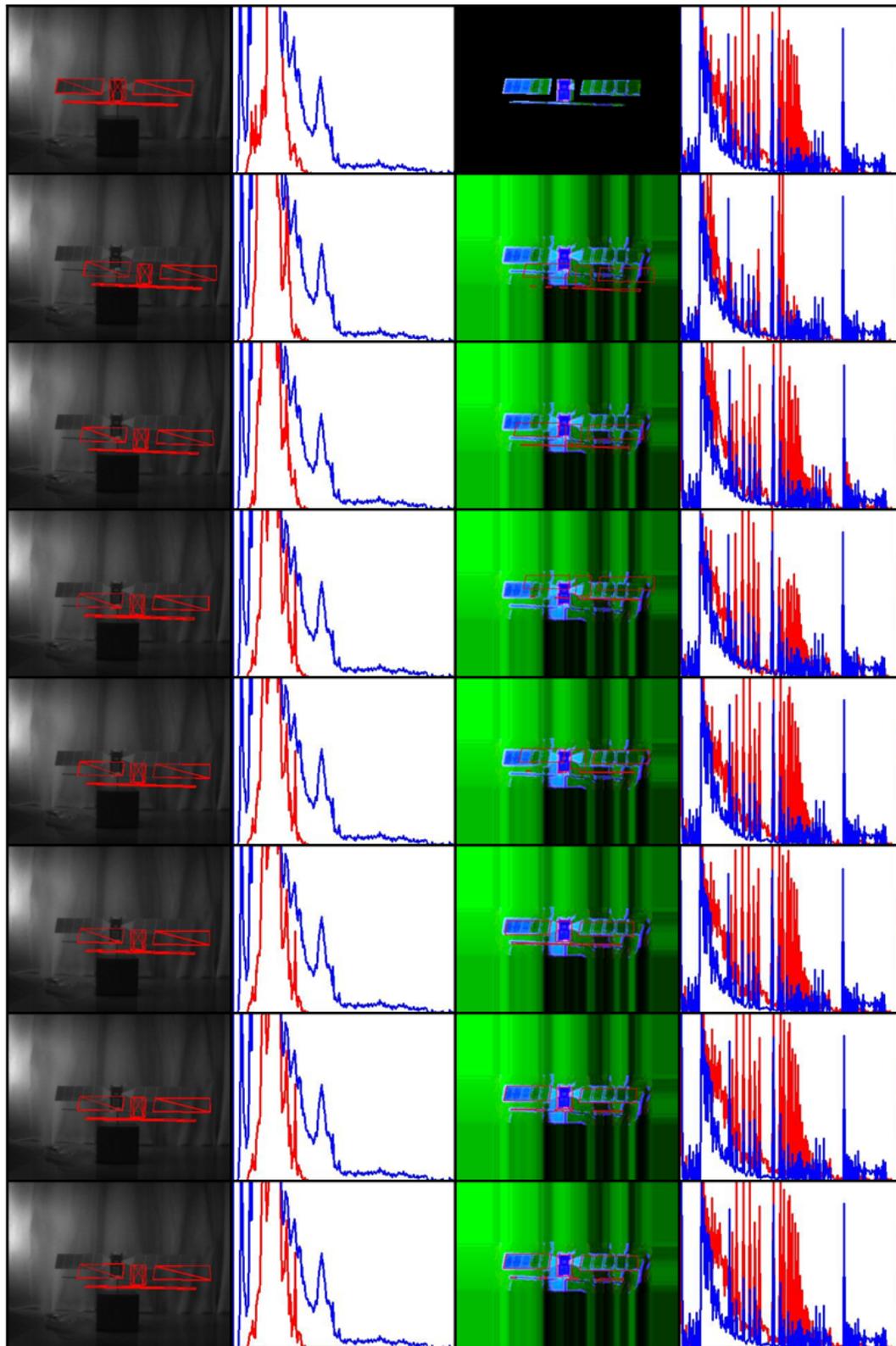


Figure 4.25: RSM infrared image pose initialisation.

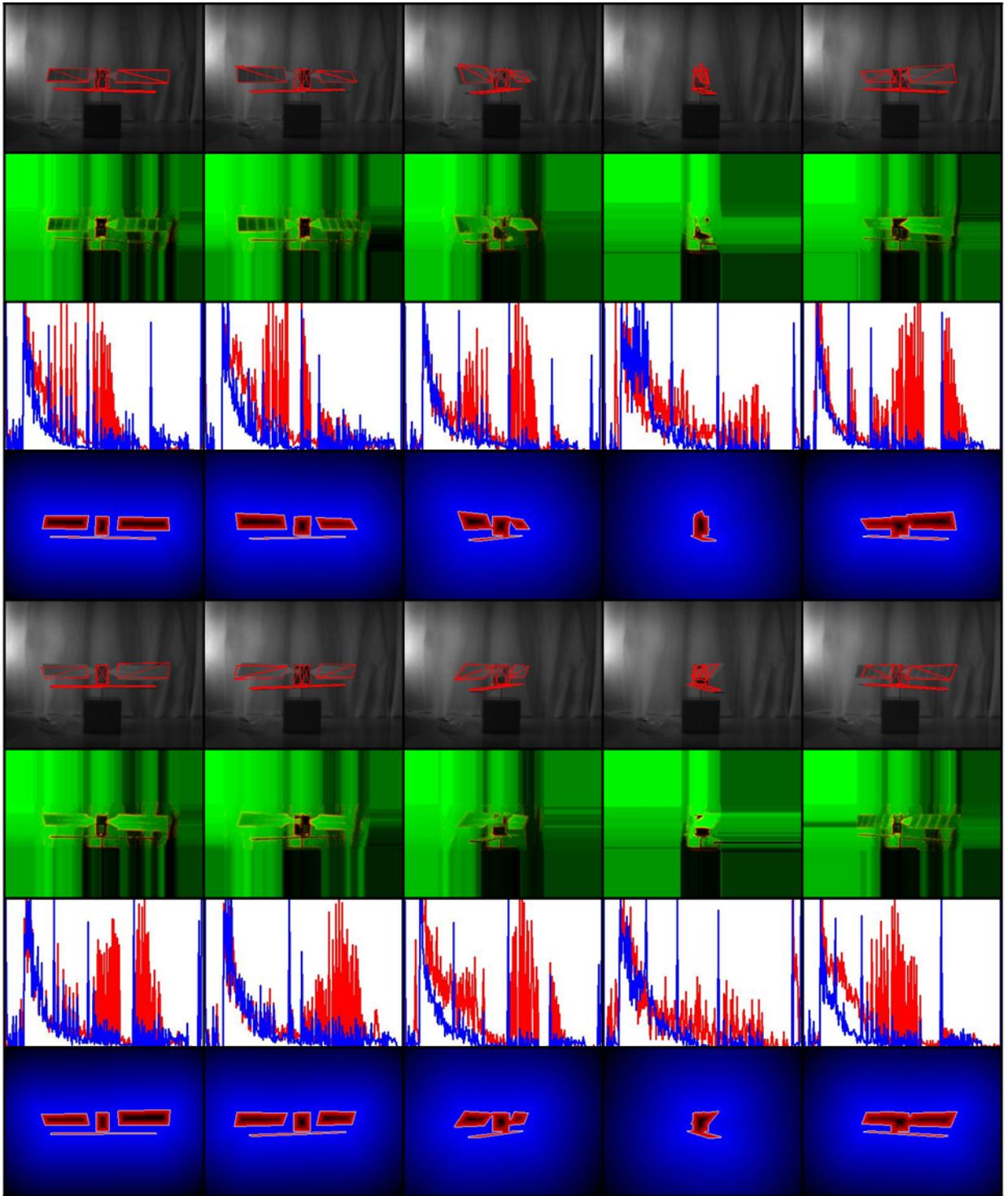


Figure 4.26: Monocular infrared image pose estimation.

Chapter 5

Appearance-based Methods

The appearance-based method for pose estimation directly matches the captured image with an internally stored image tagged with pose labels. The memory storage and search computation requirements are too large to accommodate all changes in scale, rotation, affine-viewpoint, illumination, image blur, occlusion, and background clutter. Fortunately, affine perspective, occlusion, and background clutter do not affect some spacecraft ProxOps scenarios. Occlusion due to frame crop and background clutter is irrelevant in cases where the spacecraft can be in full view with black space as the background. Affine perspective change is limited for mid to far target views, where the spacecraft occupies half to one-quarter of the image. In this ProxOps scenario, the camera can use a small FoV and a larger focal length to minimise affine perspective distortions. This chapter provides methods to efficiently handle scale, in-plane and out-of-plane rotations, illumination, image blur, and occlusion by shadowing using the appearance-based method.

The Principal Component Analysis (PCA), also known as Karhunen-Loève transformation [386], is widely used in statistical analysis in various disciplines such as meteorology, biosciences, medicine, and computer vision [387]. The PCA technique is an effective tool to reduce data dimensions [388, 389], and in computer vision, PCA has a large focus in facial recognition as *eigenpictures* [390], head motion tracking by Turk and Pentland as *eigenfaces* [391], and also in facial detection [392]. A large body of research was developed to improve facial detection by enhancement of the test procedure through weighting [393], 3D face reconstruction [394, 395], robustness to illumination and view [396], generalised 2-D PCA to increase robustness [397], and the use of kernels for non-linear feature extraction [81, 386, 398–400]. The PCA was combined with ICP [401] for spacecraft pose estimation using LIDAR [402], and PCA has been used to reduce image descriptor dimensions [403]. We focus on an appearance-based algorithm that transforms into the *parametric eigenspace* to reduce the image dimensions [404]. We also implement the recently

developed Kernel PCA (KPCA) by Liwicki *et al.* [81] called Euler-PCA (*ePCA*) to resolve shadow occlusion. The *ePCA* image occlusion optimisation uses a robust dissimilarity measure based on the Euler representation of complex numbers. The PCA appearance-based pose estimation software is in Appendix C.5.

5.1 Principal Component Analysis

The PCA transforms an input image into the eigenspace and only keeps the principal dimensions while discarding the rest. The dimension reduction requires less model image memory storage and allows faster image vector matching. The PCA appearance-based pose estimation requires each image to contain only one target, the full object boundary must be in view, and the camera view is a weak perspective. As previously mentioned, for spacecraft imagery, these assumptions are reasonable at 100 to 200 meters for a 5 to 10 ton class satellite. Finally, for scale and illumination invariance, the images are centered and the pixel values are normalised to one. The processed input and internal model image both enclose the target object. The following describes PCA for 3D orientation estimation.

Given an image of \mathbf{I} with resolution $R \times C$, where the i^{th} and j^{th} pixel of \mathbf{I} is normalised to $\hat{I}_{ij} = I_{ij}/\bar{I}$, where

$$\bar{I} = \left[\sum_{i=1}^R \sum_{j=1}^C I_{ij}^2 \right]^{1/2}. \quad (5.1)$$

$\hat{\mathbf{I}}$ is resised into an array $\hat{\mathbf{x}}$, where

$$\hat{\mathbf{x}} = \left[\hat{I}_{11} \quad \hat{I}_{12} \quad \dots \quad \hat{I}_{1C} \quad \hat{I}_{21} \quad \hat{I}_{22} \quad \dots \quad \hat{I}_{RC} \right]^T. \quad (5.2)$$

The dimension of $\hat{\mathbf{x}}$ is the resolution of the image $N = R \times C$. Given a training image sequence of M images, the mean value of $\hat{\mathbf{x}}$ is

$$\bar{\mathbf{x}} = \frac{1}{N} \sum_{j=1}^N \hat{\mathbf{x}}_j. \quad (5.3)$$

The training data matrix \mathbf{X} is formed as a $N \times M$ matrix where $\mathbf{X} \in \mathbb{R}^{N \times M}$

$$\mathbf{X} = \hat{\mathbf{X}} - \bar{\mathbf{x}}\mathbf{1}^T = \left[\hat{\mathbf{x}}_1 \quad \dots \quad \hat{\mathbf{x}}_M \right] - \bar{\mathbf{x}}\mathbf{1}^T, \quad (5.4)$$

where $\mathbf{1}$ is an array of ones of the size $M \times 1$. The covariance matrix of \mathbf{X} is

$$\bar{\mathbf{Q}} = \frac{\mathbf{Q}}{M-1} = \frac{\mathbf{X}\mathbf{X}^T}{M-1}. \quad (5.5)$$

Note $\bar{\mathbf{Q}}$ and \mathbf{Q} has the same eigenvectors and therefore it is more common to use \mathbf{Q} for direct PCA, and $\mathbf{Q} \in \mathbb{R}^{N \times N}$. The covariance matrix \mathbf{Q} can be transformed into the eigenspace, by

$$\Lambda = \mathbf{P}^T \mathbf{Q} \mathbf{P}, \quad (5.6)$$

where \mathbf{P} is the matrix of eigenvectors, $\mathbf{P} \in \mathbb{R}^{N \times N}$

$$\mathbf{P} = \begin{bmatrix} \hat{\mathbf{e}}_1 & \hat{\mathbf{e}}_2 & \dots & \hat{\mathbf{e}}_N \end{bmatrix}. \quad (5.7)$$

By *Eigenspace Representation Theorem* [404], the j^{th} normalised image array (columns from the full image matrix) can be reconstructed by a linear combination of the eigenspace unit vectors.

$$\hat{\mathbf{x}}_j = \bar{\mathbf{x}} + \sum_{i=1}^N g_{ji} \hat{\mathbf{e}}_i, \quad (5.8)$$

Proof

Transform \mathbf{X} from Eq. (5.4) by applying the eigenspace transformation matrix \mathbf{P}^T ,

$$\mathbf{G} = \mathbf{P}^T \mathbf{X}, \quad (5.9)$$

hence,

$$\hat{\mathbf{X}} = \bar{\mathbf{x}} \mathbf{1}^T + \mathbf{P} \mathbf{G}, \quad (5.10)$$

where $\mathbf{g}_j \in \mathbb{R}^N$ and $\mathbf{G} \in \mathbb{R}^{N \times M}$,

$$\mathbf{G} = \begin{bmatrix} \mathbf{g}_1 & \mathbf{g}_2 & \dots & \mathbf{g}_M \end{bmatrix}. \quad (5.11)$$

The individual \mathbf{g}_j is therefore the columns of \mathbf{X} transformed into the eigenspace. ■

Consider the comparison of two images, n and m . The image correlation function of

two images $\hat{\mathbf{I}}_n$ and $\hat{\mathbf{I}}_m$, is defined by the symbol \bullet ,

$$c = \hat{\mathbf{I}}_n \bullet \hat{\mathbf{I}}_m \triangleq \sum_{i=1}^R \sum_{j=1}^C \hat{\mathbf{I}}_n(i, j) \hat{\mathbf{I}}_m(i, j), \quad (5.12)$$

the larger the value c , the more similar $\hat{\mathbf{I}}_n$ and $\hat{\mathbf{I}}_m$ is to each other. Equation 5.12 can be expressed by $\hat{\mathbf{x}}$ from Eq. (5.2) as,

$$c = \hat{\mathbf{I}}_n \bullet \hat{\mathbf{I}}_m = \langle \hat{\mathbf{x}}_n, \hat{\mathbf{x}}_m \rangle = \hat{\mathbf{x}}_n^T \hat{\mathbf{x}}_m, \quad (5.13)$$

where $\langle \cdot, \cdot \rangle$ is the inner dot product. From Eq. (5.13), the Euclidean distance between the two normalised images can be related to the image correlation by

$$c = 1 - \frac{1}{2} \|\hat{\mathbf{x}}_n - \hat{\mathbf{x}}_m\|_2^2, \quad (5.14)$$

where c is between 0 and 1, such that the maximum correlation is the minimum Euclidean distance.

Proof

The norm of $\hat{\mathbf{x}}_n$ and $\hat{\mathbf{x}}_m$ is 1, and therefore,

$$\|\hat{\mathbf{x}}_n - \hat{\mathbf{x}}_m\|_2^2 = 2 \left(1 - \hat{\mathbf{x}}_n^T \hat{\mathbf{x}}_m \right), \quad (5.15)$$

Equation (5.14) can be derived by combining Eq. (5.13) and Eq. (5.15). ■

Since the eigenspace \mathbf{g}_j is a rotation transformation of $\hat{\mathbf{x}}_j$ per Eq. (5.9), then the Euclidean distance of eigenspace vectors is the Euclidean distance between the two images.

$$\|\mathbf{g}_n - \mathbf{g}_m\|_2^2 = \|\hat{\mathbf{x}}_n - \hat{\mathbf{x}}_m\|_2^2. \quad (5.16)$$

5.1.1 Dimensional Reduction

The advantage of converting the image data into eigenspace is the ability to reduce the so-called *curse of dimensionality*. Examining Eq. (5.8), only the *principal components* of \mathbf{g}_j contains majority of the image information, the smaller insignificant terms can therefore be discarded as they will not affect the summation. When computing Eq. (5.6), the eigenvalue

is ordered from large to small, this also order the eigenspace transformation matrix \mathbf{P} and the components in the vector \mathbf{g}_j . Equation 5.8 can be represented to dimension K instead of N where $N \gg K$. The new smaller \mathbf{P} and \mathbf{G} matrices are denoted by $\tilde{\mathbf{P}}$ and $\tilde{\mathbf{G}}$ as follows,

$$\tilde{\mathbf{P}} = \begin{bmatrix} \hat{\mathbf{e}}_1 & \hat{\mathbf{e}}_2 & \dots & \hat{\mathbf{e}}_K \end{bmatrix} \quad (5.17)$$

$$\tilde{\mathbf{G}} = \begin{bmatrix} \tilde{\mathbf{g}}_1 & \tilde{\mathbf{g}}_2 & \dots & \tilde{\mathbf{g}}_M \end{bmatrix} \quad (5.18)$$

where $\tilde{\mathbf{P}} \in \mathbb{R}^{N \times K}$, $\tilde{\mathbf{g}}_j \in \mathbb{R}^K$, and $\tilde{\mathbf{G}} \in \mathbb{R}^{K \times M}$. Experiments show the K dimension can be as low as 10 where as N is 76,800 for a low resolution 320×240 size image, which is a significant reduction in processing time.

5.1.2 Low-rank Matrix Approximation

While Eq. (5.6) appears to be elegant, \mathbf{P} cannot be quickly solved in practice as a $N \times N$ eigenspace basis vector given the enormous size of N . A low-rank matrix approximation is typically used to remedy this problem. Consider the Singular-Value-Decomposition (SVD) of \mathbf{X} as

$$\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T. \quad (5.19)$$

Substitute Eq. (5.19) into Eq. (5.5) produces

$$\mathbf{Q} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T\mathbf{V}\mathbf{\Sigma}^T\mathbf{U}^T = \mathbf{U}\mathbf{\Sigma}\mathbf{\Sigma}^T\mathbf{U}^T = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T. \quad (5.20)$$

Compare Eq. (5.20) with Eq. (5.6), \mathbf{U} is the eigenspace vector \mathbf{P} , and $\mathbf{\Sigma}$ is the square root of the eigenvalue matrix $\mathbf{\Lambda}$ assuming $M \geq N$. Substituting Eq. (5.19) into Eq. (5.9) produces

$$\mathbf{G} = \mathbf{P}^T\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T = \mathbf{\Sigma}\mathbf{V}^T. \quad (5.21)$$

The advantage of PCA is the ability to use only a very few number of principal components from the eigenspace that contains majority of the data information. Selecting the first K terms from \mathbf{P} instead of the full N^{th} dimension, Eq. (5.21) becomes

$$\tilde{\mathbf{G}} = \tilde{\mathbf{P}}^T\mathbf{X} = \tilde{\mathbf{\Sigma}}\mathbf{V}^T. \quad (5.22)$$

The low-rank matrix approximation [405] of Eq. (5.19) is therefore,

$$\mathbf{X} = \tilde{\mathbf{P}}\tilde{\Sigma}\mathbf{V}^T. \quad (5.23)$$

Equation (5.23) is solved using the `svds()` function from MATLAB [406, 407] with K value input. Some other methods for solving low-rank approximation SVD are provided by Rokhlin *et al.* [408] and Halko *et al.* [409].

5.1.3 Inference

Given a test image captured during operations, it is first converted into an image array $\hat{\mathbf{x}}_t$ by the same steps from Eq. (5.2). It is then transformed into the eigenspace by using the following,

$$\tilde{\mathbf{g}}_t = \tilde{\mathbf{P}}^T (\hat{\mathbf{x}}_t - \bar{\mathbf{x}}) \quad (5.24)$$

where $\tilde{\mathbf{g}}_t \in \mathbb{R}^K$. This is compared with all vector columns in the $\tilde{\mathbf{G}}$ matrix to determine the minimum Euclidean distance match $\tilde{\mathbf{g}}_\wedge \in \mathbb{R}^K$.

$$\tilde{\mathbf{g}}_\wedge = \arg \min_{\tilde{\mathbf{g}}_j \in \tilde{\mathbf{G}}} \|\tilde{\mathbf{g}}_t - \tilde{\mathbf{g}}_j\|_2^2 \quad (5.25)$$

The training image will have known pose tag associated with it. If the match falls within two nearest training images, the pose is linearly interpolated between the two poses.

5.2 Appearance Matching

The pose matching algorithm is separated in two phases: a training phase, as provided in Algorithm 8, that is computed off-line before operational use to store labelled PCA model images; and an inference phase, as provided in Algorithm 9, that matches the spacecraft pose during real-time operations. If the model image set only contains 3D rotational images, then the input image is cropped to a fully visible target, and the spacecraft position must be computed outside the appearance matching approach.

Algorithm 8 PCA_TRAIN

```

1: procedure COMPUTE_PCA_DATABASE( $\mathbf{I}_j (j = 1, \dots, M)$ ,  $K$ )
2:   Generate  $M$  images by rotating the CS CAD model.
3:   Tag every image  $\mathbf{I}_j$  with the associated pose.
4:   for all  $M$  images in the dataset. do
5:     Segment CS foreground using methods from Chap. 4 Sec. 4.2.
6:     Crop to a fully visible target that does not touch the edge of the image.
7:     Normalise pixel intensity using Eq. (5.1).
8:     Convert the normalised image into  $\hat{\mathbf{x}}$  representation by Eq. (5.2).
9:   Compute average image vector  $\bar{\mathbf{x}}$  of the entire database by Eq. (5.3).
10:  Form the image database  $\mathbf{X}$  in accordance to Eq. (5.5).
11:  Perform low-rank matrix approximation SVD by Eq. (5.19),  $\tilde{\mathbf{P}} = svds(\mathbf{X}, K)$ .
12:  Compute  $\tilde{\mathbf{G}}$  by transforming image database into eigenspace by Eq. (5.22).
13:  return  $\bar{\mathbf{x}}, \tilde{\mathbf{G}}, \tilde{\mathbf{P}}$ .

```

Algorithm 9 PCA_INFERENCE

```

1: procedure PCA_POSE_RECOGNITION( $\mathbf{I}_t, \bar{\mathbf{x}}, \tilde{\mathbf{G}}, \tilde{\mathbf{P}}$ )
2:   Perform steps 5 to 7 from Algorithm 8 on captured image  $\mathbf{I}_t$  and form  $\hat{\mathbf{x}}_t$ .
3:   Compute eigenspace representation of captured image  $\tilde{\mathbf{g}}_t$  using Eq. (5.24).
4:   Find minimum matching  $\tilde{\mathbf{g}}_j$  in accordance to Eq. (5.25).
5:   Compare dataset image  $j + 1$  and  $j - 1$  for second minimum match.
6:   Interpolate rotation ( $\mathbf{R}$ ) between minimum and second minimum match.
7:   if Position label is available and input image is not cropped. then
8:     Interpolate translation ( $\mathbf{t}$ ) between minimum and second minimum match.
9:   return  $\mathbf{t}, \mathbf{R}$ .

```

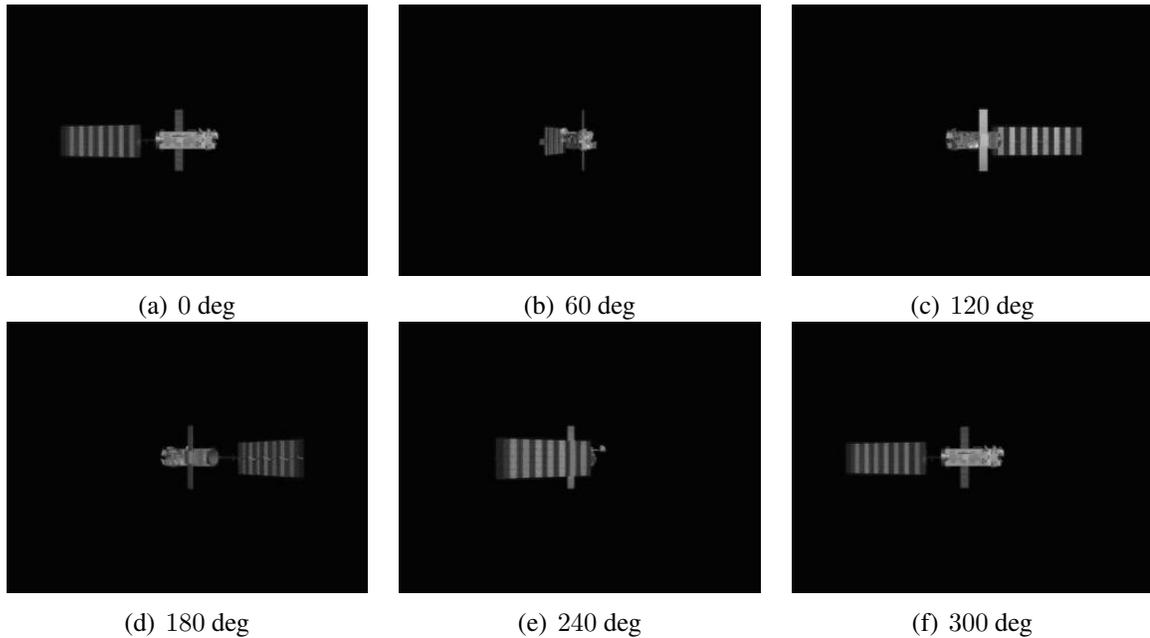


Figure 5.1: Envisat pitch-axis rotation images.

5.3 Pose Estimation Performance

This section provides PCA pose estimation results and discussions for synthetic and hardware thermal camera images for single axis and 3D target motions. All simulations are performed on Intel® Core™ 2 Quad Q6600 – 2.4GHz Processor running on 32bit Windows-Vista-SP2.

5.3.1 Single-axis Pose Estimation

The single-axis motion example rotates a synthetic 3D model Envisat about the pitch axis. The camera resolution is 320×240 , with focal length and scale factor 439.967, 432.427, and -0.070 for x , y , and θ axis respectively. The Envisat pitch axis motion sequence is provided in Fig. 5.1. The total number of training frames, M , and the maximum principal components, K , are two main factors affecting the pose estimation precision. In the former, increasing the total number of training images will increase estimation precision by reducing ambiguity between similar pose. The trade-off is, with increasing training set, the search time will also increase. As pose degrees-of-freedom (DOF) increase, the search order of magnitude will increase by the total training frame per axis to the power of

the pose DOF. The PCA pose estimation results for increasing number of training frames are provided in Fig. 5.2 for M equal to 10 to 300 frames respectively. Interpolation after matching provides better precision, this is especially important if there are low number of training images. To interpolate the pose results, two nearest matches are selected after an angular tolerance thresholding. The angular tolerance is set on the order of the pose resolution based on number of training images used. For example, an angular tolerance of 45 degrees is used for the minimum M of 10 training frames spanning 360 degrees. Figures 5.2(a) and 5.2(b) shows estimation without and with interpolation. If interpolation is not performed, the pose value jumps from closest frames and resulting in a step pattern. The interpolation caused an overhead on the computation that is 0.7% increase to the total processing time for all 300 test frames. Furthermore, when the training set is coarse, it is more likely to result in wrong matches in a mirror angle. This is evident in Fig. 5.2(a), at every 30 degrees, the choices of two neighbours, -30 and $+30$ degrees, selected the wrong choice and caused the interpolation to run backwards. This error was corrected with increasing M Figs. 5.2(b) to (f). Figure 5.3 and Table 5.1 shows the timing and error metric for the single axis Envisat example. The average single image testing time is the time it takes to search the database of principal components training image. This time will increase linearly with increasing test images; for $M = 300$, the maximum search time is 14 ms. Outliers may cause wrong matches and can be removed using an angle tolerance. One common cause for outlier is the 360 deg wrapping; it will result in the maximum error ceiling for all M image trials. Figure 5.3 shows a large reduction in maximum angular error from 37 degrees to 5 degrees when number of training images is increased from 10 to 30 frames, a decrease of 85%; this error reduction has a timing increase of 7%.

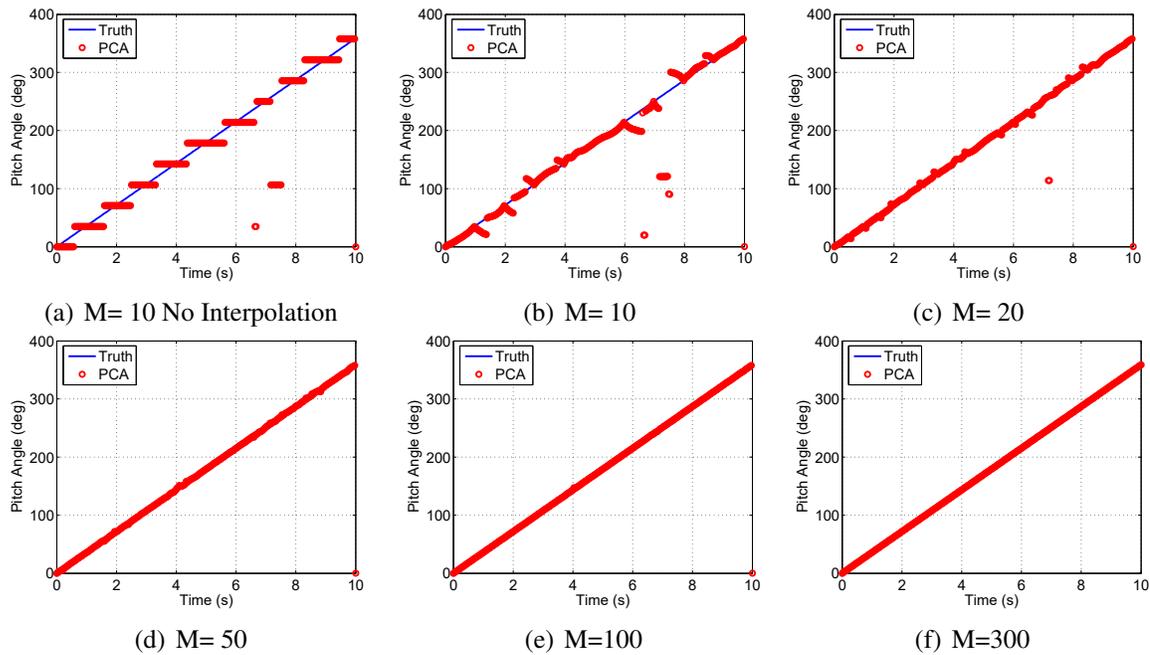
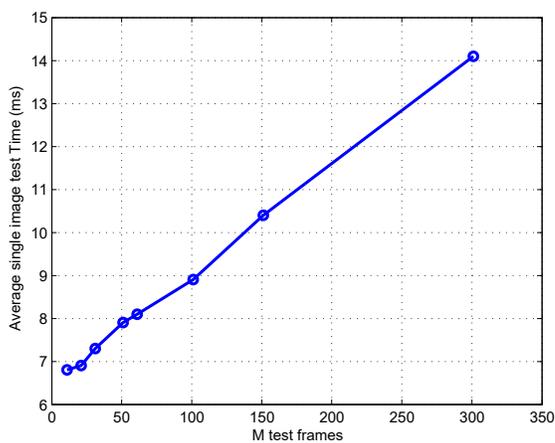


Figure 5.2: Pitch axis pose estimation vs. ground truth.

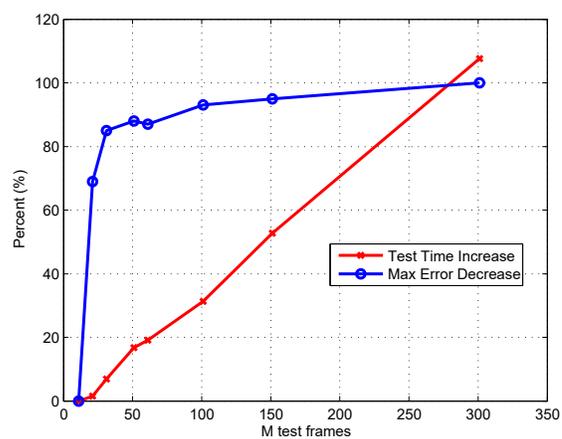
Table 5.1: Variations in number of training frames.

Number of Training Frames	Avg Est Time per Frame (ms)	Est. Time Percent Increase (%)	Mean Ang Err (deg)	Max Ang Err (deg)	Max Err Percent Decrease (%)
10	6.8	0.0	6.378	37.320	0
20	6.9	1.5	1.553	11.421	69
30	7.3	6.9	0.712	5.626	85
50	7.9	16.8	0.355	4.398	88
60	8.1	19.1	0.290	4.709	87
100	8.9	31.3	0.086	2.642	93
150	10.4	52.7	0.047	1.945	95
300	14.1	107.6	0.000	0.000	100

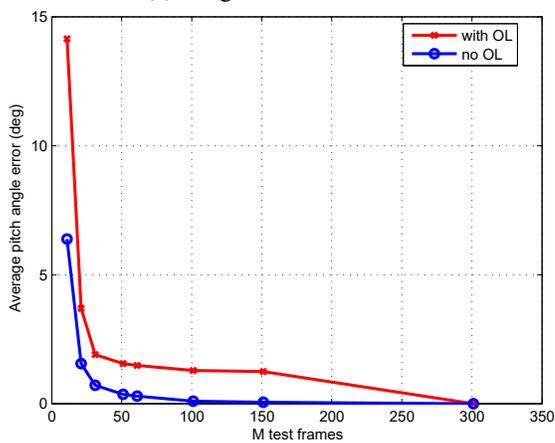
The number of principal components used to describe the image, K , is also an important factor in the PCA pose estimation precision. With increasing K , the SVD computation and the training time increases. Figure 5.4(a) shows a linear increase in the training and inference time with increasing number of principal components, where the evaluation in Fig. 5.4(b) used five cases with increasing number of training images from 10 to 100. A



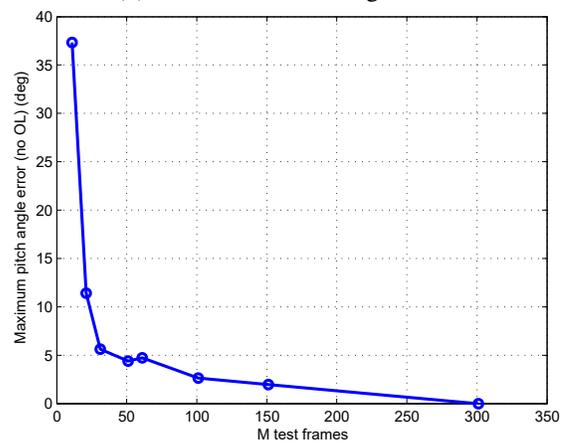
(a) Single frame test time



(b) Percent time and angle error



(c) Average pitch angle error



(d) Maximum pitch angle error

Figure 5.3: Testing metric resulting from M number of training images.

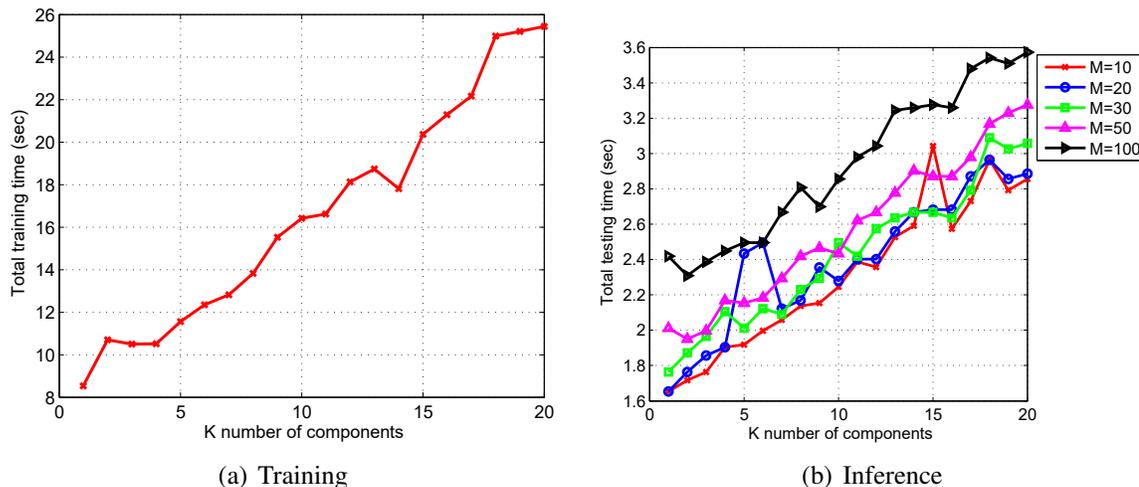


Figure 5.4: Computation time resulting from K components.

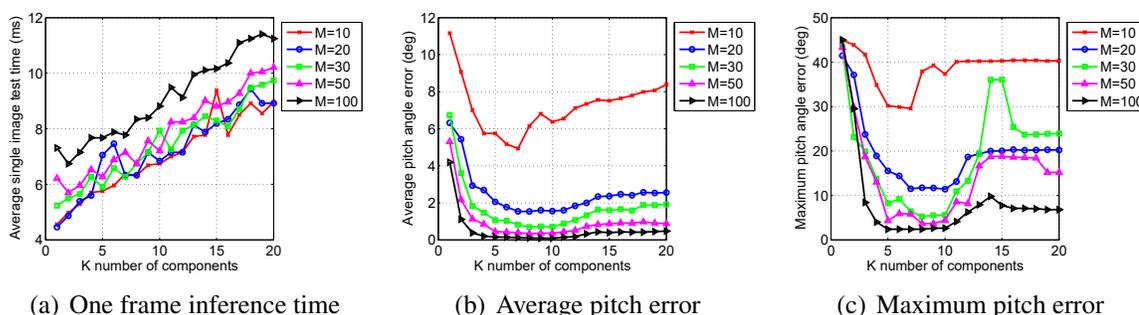


Figure 5.5: Inference metric resulting from K components.

noticeable increase in total inference time occurred as total training images increased from 10 to 100. Figure 5.5 shows the timing and pose error for the inference phase with K equaling to 1 to 20 principal components. Five datasets of M equal to 10, 20, 30, 50, and 100 were compared. A large jump occurred in average single image inference time when the training set is $M = 100$ images; whereas from $M = 50$ and below, the inference timing remains relatively the same. All inference durations are linearly increasing with the number of principal components. The average pitch angle and the maximum pitch angle both show a dip in precision between 5 to 10 principal components, the optimal number of components is 8. With an increasing number of components, angular error settles to a constant value; because the usefulness of the higher eigen-dimension information diminishes. Finally, the pitch error is significantly higher when the inference dataset is only 10 images.

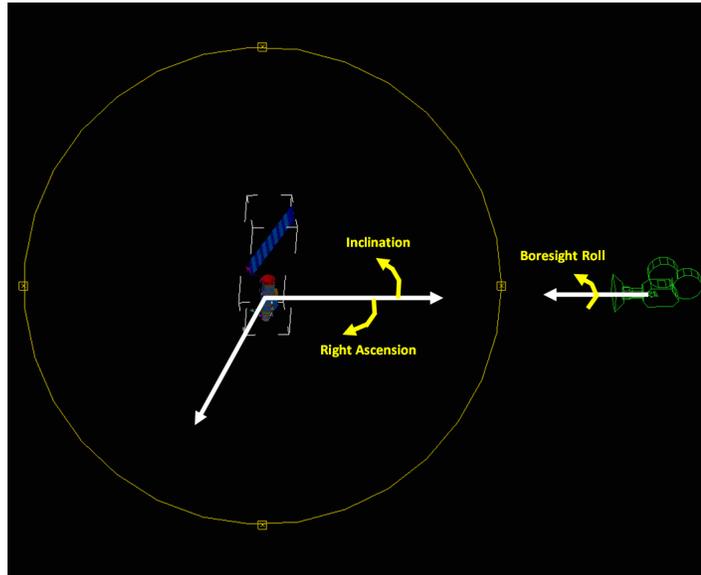


Figure 5.6: 3D environment and coordinate system for training image generation.

5.3.2 Multi-axis Pose Estimation

In the multi-axis pose estimation case, the target spacecraft or CS, is allow to rotate about the X , Y , and Z axis. The CS rotation is equivalent to revolving the camera about a stationary target, which is much easier to implement in the 3D environment. We lock the camera boresight onto the CS Center-of-Mass (CoM) and keep the relative distance between the CS and the SS camera constant. Next, we use spherical coordinates with inclination and right ascension as shown in Fig. 5.6 to trace out a spherical path about the target. The spherical coordinate and the boresight roll angle is kinematically transformed into quaternions. Based on the findings of Sec. 5.3.1, 30 intervals were used for each of the inclination, right ascension, and boresight roll axis. At the north and south poles, the redundant right ascension frames were removed. The final number of training images is $M = 12,660$ that covers the entire rotation space. We rescaled [15] the original image to $1/2$, $1/4$, $1/8$, and $1/16$ size for manageable SVD computation. The rescaling removes every other line of pixels. Figure 5.7 provides the four training resolution for the same image. Figure 5.10(a) shows the eigenspace error norm between the inference and training images. The low-resolution case has a higher scatter, which indicates the PCA method is sensitive to image resolution. Figure 5.10(b) shows the training image resolution has little influence on the matching duration. The mean search time is approximately 62.8 ms. The



Figure 5.7: Reduction in image resolution using pyramids

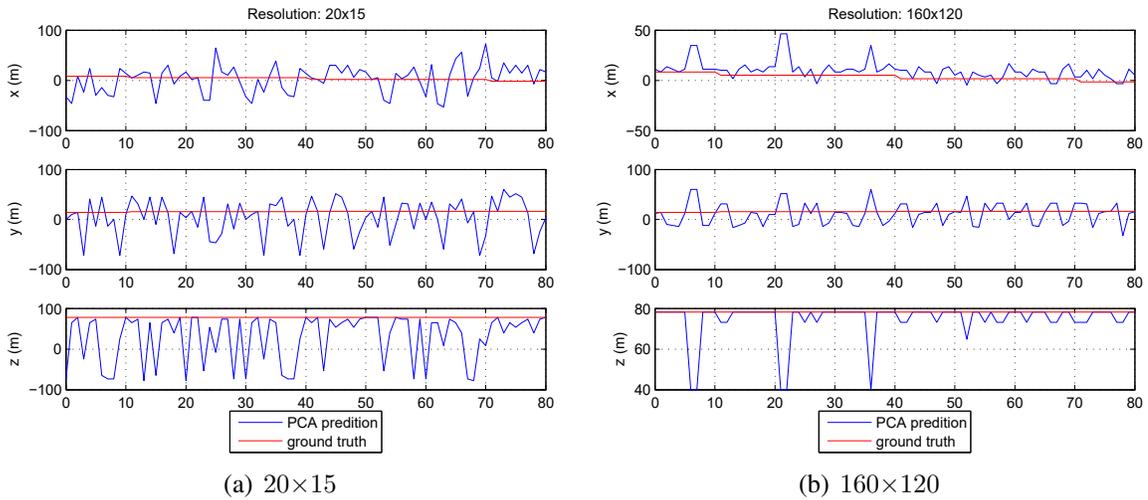


Figure 5.8: Estimated position from different training resolutions.

PCA is faster than the iterative point-based method per Chapter 6.

5.4 Image Occlusion Optimisation

On Earth, sunlight is scattered in all directions by the atmosphere; lighting in space, on the other hand, usually comes from one direction and causes darker and harder shadowing. Shadow occlusion is difficult to avoid; an external remedy is to include anticipated shadowing in the stored model images; however, this approach could lead to more errors if the lighting direction assumptions are wrong. Alternatively, we improve image robustness using PCA-based extraction to handle occlusion by applying non-linear feature extracting Kernel PCA (KPCA) and operating in high-dimensional feature spaces. The KPCA use pre-images [398, 399] and is developed as part of SVM [121, 410, 411]. The kernel machine maps the input space \hat{x} into a higher dimensional feature space or *reproducing kernel Hilbert space* (rkHs), \mathcal{H} , using a non-linear function $\phi(\hat{x}_j)$. The so called *kernel trick* exploits the similarity measure between images by using the inner dot product [386]. In

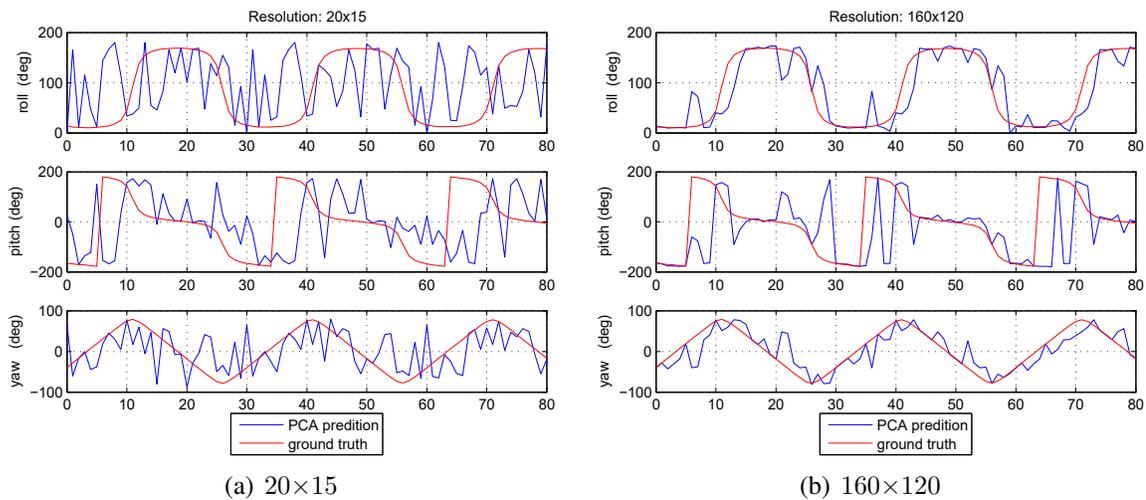


Figure 5.9: Estimated orientation from different training resolutions.

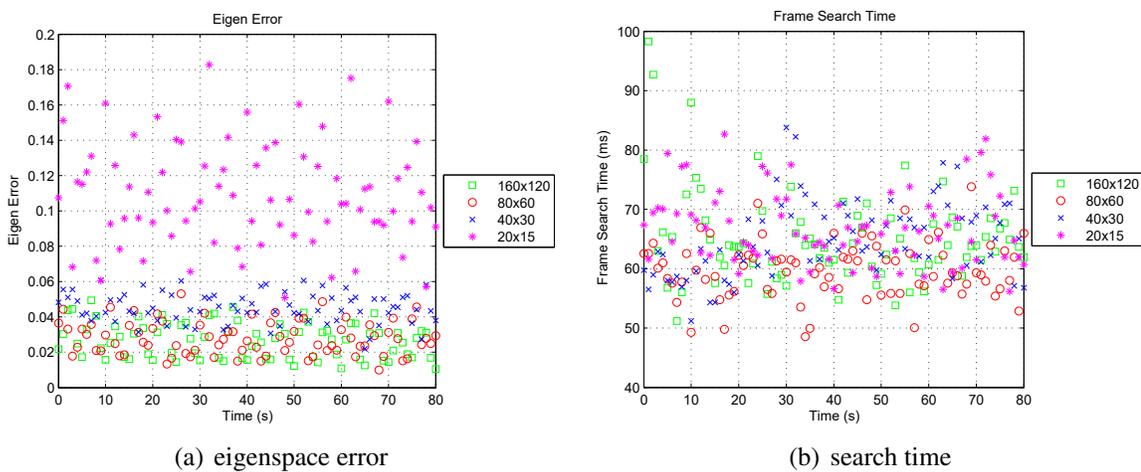


Figure 5.10: Eigenspace error and test search time.

general, the formula for kernel κ in the rkHs is

$$\kappa(\hat{\mathbf{x}}_n, \hat{\mathbf{x}}_m) = \langle \phi(\hat{\mathbf{x}}_n), \phi(\hat{\mathbf{x}}_m) \rangle_{\mathcal{H}}. \quad (5.26)$$

The objective is to improve the pose estimation by image matching and solving an optimization problem [399] such that,

$$\hat{\mathbf{x}}_{\wedge} = \arg \min_{\hat{\mathbf{x}}_l \in \mathbf{X}} \|\phi(\hat{\mathbf{x}}_l) - \beta \beta^T \phi(\hat{\mathbf{x}}_j)\|_2^2, \quad (5.27)$$

where β is a linear subspace orthonormal basis in rkHs. Some commonly used Kernels that are projective, such as polynomial and exponential, or radial, such as Gaussian and Laplacian, can be found in a survey of KPCA by Honeine and Richard [386]. The Euler-PCA (*ePCA*) is a KPCA based on the Euler representation of complex numbers introduced by Liwicki *et al.* [81]. In *ePCA* rkHs, the L_2 -norm can be related to a dissimilarity measure originally introduced by Fitch *et al.* [412], refer to Sec. 5.4.1; this effectively provide a fast and statistically more robust estimation. In *ePCA*, the normalised image $\hat{\mathbf{x}}$ is mapped onto a complex number representation $\mathbf{z}_j \in \mathbb{C}^N$

$$\mathbf{z}_j = \frac{1}{\sqrt{2}} e^{i\alpha\pi\hat{\mathbf{x}}_j} = \frac{1}{\sqrt{2}} (\cos(\alpha\pi\hat{\mathbf{x}}_j) + i \sin(\alpha\pi\hat{\mathbf{x}}_j)). \quad (5.28)$$

Grouping \mathbf{z}_j gives the data matrix $\mathbf{Z} \in \mathbb{C}^{N \times M}$ in rkHs

$$\mathbf{Z} = \begin{bmatrix} \mathbf{z}_1 & \mathbf{z}_2 & \dots & \mathbf{z}_M \end{bmatrix}. \quad (5.29)$$

The Hermitian kernel κ for *ePCA* is represented by $\mathbf{K} \in \mathbb{C}^{M \times M}$ and can then be formulated as

$$\mathbf{K} = \mathbf{Z}^H \mathbf{Z}, \quad (5.30)$$

where $(\cdot)^H$ is the complex conjugate transposition of a matrix. The eigen decomposition of Eq. (5.30) is

$$\Lambda_c = \mathbf{P}_c^H \mathbf{K} \mathbf{P}_c, \quad (5.31)$$

where $\mathbf{P}_c \in \mathbb{C}^{M \times M}$. Using similar method as Eq. (5.23), the lower dimension $\tilde{\mathbf{P}}_c \in \mathbb{C}^{M \times R}$ is found,

$$\mathbf{K} = \tilde{\mathbf{P}}_c \tilde{\Sigma}_c \mathbf{V}_c^H. \quad (5.32)$$

Using $\tilde{\mathbf{P}}_c$ in Eq. (5.31), the lower dimension $\tilde{\Lambda}_c \in \mathbb{C}^{R \times R}$ can be found

$$\tilde{\Lambda}_c = \tilde{\mathbf{P}}_c^H \mathbf{K} \tilde{\mathbf{P}}_c. \quad (5.33)$$

For $M \geq R$, $\tilde{\Sigma}_c$ and $\tilde{\Lambda}_c^{1/2}$ contains the same eigenvalues. The basis mapping β for *ePCA* is represented by $\tilde{\mathbf{B}} \in \mathbb{C}^{N \times R}$ and is formulated as

$$\tilde{\mathbf{B}} = \mathbf{Z} \tilde{\mathbf{P}}_c \tilde{\Lambda}_c^{-1/2}, \quad (5.34)$$

and $\tilde{\mathbf{B}}$ is the eigenspace mapping of $\tilde{\mathbf{K}} \in \mathbb{C}^{N \times N}$, where

$$\tilde{\mathbf{K}} = \mathbf{Z} \mathbf{Z}^H. \quad (5.35)$$

Combining Eq. (5.34) and Eq. (5.35), we can also show $\tilde{\mathbf{B}}$ is also an eigenspace basis vectrix,

$$\begin{aligned} \tilde{\mathbf{B}}^H \tilde{\mathbf{K}} \tilde{\mathbf{B}} &= \tilde{\Lambda}_c^{-1/2} \tilde{\mathbf{P}}_c^H \mathbf{Z}^H \mathbf{Z} \mathbf{Z}^H \mathbf{Z} \tilde{\mathbf{P}}_c \tilde{\Lambda}_c^{-1/2} \\ &= \tilde{\Lambda}_c^{-1/2} \tilde{\mathbf{P}}_c^H \tilde{\mathbf{P}}_c \tilde{\Lambda}_c \tilde{\mathbf{P}}_c^H \tilde{\mathbf{P}}_c \tilde{\Lambda}_c^{-1/2} \\ &= \tilde{\Lambda}_c^{-1/2} \tilde{\Lambda}_c \tilde{\Lambda}_c^{-1/2} \\ &= \tilde{\Lambda}_c^{1/2} \tilde{\Lambda}_c^{1/2} \\ &= \tilde{\Lambda}_c, \end{aligned} \quad (5.36)$$

where Eq. (5.36) is equivalent to Eq. (5.33). Finally, the reconstruction of $\check{\mathbf{z}}_j \in \mathbb{C}^N$ is,

$$\check{\mathbf{z}}_j = \tilde{\mathbf{B}} \tilde{\mathbf{B}}^H \mathbf{z}_j. \quad (5.37)$$

In this form, we may compute matching based on dissimilarity to be discussed in the next section. The advantage of the *ePCA* is when inverting from the rkHs, or the so-called *Euler complex number feature space*, back to the image space, there is a close form transformation function. Details of the inverse transformation is provided in Section 5.4.2.

5.4.1 Dissimilarity Measure

The L_2 -norm in Eq. (5.25) is optimal for *independent and identically distributed* Gaussian noise but not robust to outliers [413–415]. Some approaches have used the L_1 -norm instead for better robustness [414, 416]. Let us define $\boldsymbol{\theta}_n \triangleq \alpha\pi\hat{\mathbf{x}}_n$. A fast robust correlation scheme by Fitch *et al.* [412] suggest the so-called *dissimilarity measure*, as follows

$$\delta \triangleq \sum_{r=1}^N (1 - \cos(\theta_l(r) - \theta_j(r))). \quad (5.38)$$

A similarity can be found by applying the L_2 -norm in the Euler complex number representation space by the following,

$$\begin{aligned} \|\mathbf{z}_l - \mathbf{z}_j\|_2^2 &= \frac{1}{2} \|\cos(\boldsymbol{\theta}_l) + i\sin(\boldsymbol{\theta}_l) - \cos(\boldsymbol{\theta}_j) - i\sin(\boldsymbol{\theta}_j)\|_2^2 \\ &= \sum_{r=1}^N e^{i(\theta_l(r) + \theta_j(r))} (1 - \cos(\theta_l(r) - \theta_j(r))). \end{aligned} \quad (5.39)$$

The optimizing of Eq. (5.39) is the *ePCA* form of Eq. (5.27); this is the basis for the pre-image computation in the next section.

5.4.2 Pre-image Computation

Typically, the method to convert from the rkHs to the optimised input image space is to perform gradient ascent. Liwicki *et al.* [81] provides the *ePCA* procedures for this optimization by reforming Eq. (5.27) in the argument of the maxima form

$$\hat{\mathbf{x}}_\wedge = \arg\max_{\hat{\mathbf{x}}_l \in \mathbf{X}} \Re \left(\phi(\hat{\mathbf{x}}_l)^H \boldsymbol{\beta} \boldsymbol{\beta}^H \phi(\hat{\mathbf{x}}_j) \right), \quad (5.40)$$

where $\Re(\cdot)$ outputs the real number of the argument. In *ePCA* form, $\boldsymbol{\beta} \triangleq \mathbf{B} = \mathbf{Z}\hat{\mathbf{B}}$, then Eq. (5.40) becomes

$$\hat{\mathbf{x}}_\wedge = \arg\max_{\hat{\mathbf{x}}_l \in \mathbf{X}} \Re \left(\mathbf{z}(\hat{\mathbf{x}}_l)^H \mathbf{Z}(\mathbf{X}) \hat{\mathbf{B}} \hat{\mathbf{B}}^H \mathbf{Z}(\mathbf{X})^H \mathbf{z}(\hat{\mathbf{x}}_j) \right). \quad (5.41)$$

Applying the kernel trick [386], Eq. (5.41) becomes

$$\begin{aligned}\hat{\mathbf{x}}_\wedge &= \operatorname{argmax}_{\hat{\mathbf{x}}_l \in \mathbf{X}} \Re \left(\left[\kappa(\hat{\mathbf{x}}_l, \hat{\mathbf{x}}_1) \dots \kappa(\hat{\mathbf{x}}_l, \hat{\mathbf{x}}_M) \right] \underbrace{\hat{\mathbf{B}} \hat{\mathbf{B}}^H}_{\triangleq \mathbf{t}} \left[\kappa(\hat{\mathbf{x}}_j, \hat{\mathbf{x}}_1) \dots \kappa(\hat{\mathbf{x}}_j, \hat{\mathbf{x}}_M) \right]^H \right) \\ &= \operatorname{argmax}_{\hat{\mathbf{x}}_l \in \mathbf{X}} f(\hat{\mathbf{x}}_l),\end{aligned}\quad (5.42)$$

to maximise Eq. (5.42), the standard Newton's method is given as

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_{k-1} + \nabla f(\hat{\mathbf{x}}_{k-1}). \quad (5.43)$$

The number of steps to complete Eq. (5.43) is denoted as h . For completeness, the gradient of f is given as

$$\nabla f(\hat{\mathbf{x}}_l) = -\Im(\mathbf{H}) \Re(\mathbf{t}) + \Re(\mathbf{H}) \Im(\mathbf{t}), \quad (5.44)$$

where $\nabla(\cdot)$ is the gradient operator, $\Im(\cdot)$ is the imaginary portion of the argument, and $\mathbf{H} \in \mathbb{C}^{N \times M}$,

$$\mathbf{H} = \bar{\mathbf{z}}_l \mathbf{1}^T \circ \mathbf{Z}, \quad (5.45)$$

where $\bar{\mathbf{z}}_l$ is the complex conjugate of \mathbf{z}_l , and \circ is the Hadamard point-wise multiplication operator. Equations (5.42) to (5.45) will optimise the input when mapping the feature space image back to input space. The drawback is, the computation for the entire training set is costly, on the order of $\mathcal{O}(hNM^2)$. The advantage of using the e PCA feature space is the ability to optimally transform back to the pixel domain by a close formulation base simply on the angle of the complex number $\check{\mathbf{Z}}$,

$$\check{\mathbf{x}}_j = \frac{\angle \check{\mathbf{z}}_j}{\alpha \pi}, \quad (5.46)$$

where $\angle(\cdot)$ computes the angle of the complex number. For the same value of $\hat{\mathbf{x}}_l = \hat{\mathbf{x}}_j = \hat{\mathbf{x}}$, the rkHs representation is $\phi(\hat{\mathbf{x}}) = \mathbf{z}$.

Proof

The optimality can be shown by combining Eq. (5.37), Eq. (5.46), and Eq. (5.27).

$$\begin{aligned}
\hat{\mathbf{x}}_{\wedge} &= \arg \min_{\hat{\mathbf{x}}_l \in \mathbf{X}} \left\| \frac{1}{\sqrt{2}} e^{i\angle(\mathbf{B}\mathbf{B}^H \mathbf{z})} - \mathbf{B}\mathbf{B}^H \mathbf{z} \right\|_2^2 \\
&= \arg \min_{\hat{\mathbf{x}}_l \in \mathbf{X}} \sum_{r=1}^N \left(\frac{1}{\sqrt{2}} e^{i\angle(\mathbf{B}_r \mathbf{B}_r^H z_r)} - R(\mathbf{B}_r \mathbf{B}_r^H z_r) e^{i\angle(\mathbf{B}_r \mathbf{B}_r^H z_r)} \right)^2 \\
&= \arg \min_{\hat{\mathbf{x}}_l \in \mathbf{X}} \sum_{r=1}^N \left(\frac{1}{\sqrt{2}} - R(\mathbf{B}_r \mathbf{B}_r^H z_r) \right)^2 \\
&= \arg \min_{\hat{\mathbf{x}}_l \in \mathbf{X}} \left\| \frac{1}{\sqrt{2}} \mathbf{1} - R(\mathbf{B}\mathbf{B}^H \mathbf{z}) \right\|_2^2
\end{aligned} \tag{5.47}$$

where $R(\cdot)$ is the length of the complex number *phasor*, and $\mathbf{1}$ is the array of ones with dimension $N \times 1$. For all image in data bin, replace \mathbf{z}_j with \mathbf{Z} from Eq. (5.29), and substitute Eq. (5.34),

$$\begin{aligned}
R(\mathbf{B}\mathbf{B}^H \mathbf{Z}) &= R(\mathbf{Z} \tilde{\mathbf{P}}_c \tilde{\Lambda}_c^{-1/2} \tilde{\Lambda}_c^{-1/2} \tilde{\mathbf{P}}_c^H \mathbf{Z}^H \mathbf{Z}) \\
&= R(\mathbf{Z} \tilde{\mathbf{P}}_c \tilde{\Lambda}_c^{-1} \tilde{\mathbf{P}}_c^H \mathbf{Z}^H \mathbf{Z}) \\
&= R(\mathbf{Z} (\tilde{\mathbf{P}}_c \tilde{\Lambda}_c \tilde{\mathbf{P}}_c^H)^{-1} \mathbf{K}) \\
&= R(\mathbf{Z} \mathbf{K}^{-1} \mathbf{K}) \\
&= R(\mathbf{Z}) \\
&= \sqrt{\Re(\mathbf{Z})^2 + \Im(\mathbf{Z})^2} \\
&= \sqrt{\left(\frac{\cos \Theta}{\sqrt{2}}\right)^2 + \left(\frac{\sin \Theta}{\sqrt{2}}\right)^2} \\
&= \frac{1}{\sqrt{2}} \mathbf{1}
\end{aligned} \tag{5.48}$$

where $\mathbf{1}$ is a matrix of ones of the dimension $N \times M$ and $\Theta \in \mathbb{R}^{N \times M}$ is

$$\Theta = [\theta_1 \dots \theta_M] = \alpha\pi [\hat{\mathbf{x}}_1 \dots \hat{\mathbf{x}}_M] = \alpha\pi \hat{\mathbf{X}} \tag{5.49}$$

Compare Eq. (5.48) and Eq. (5.47), the minimum error distance is always zero for the same input image vector. Hence, Eq. (5.46) is the exact solution for the approximate optimal inverse mapping. ■

5.4.3 *e*PCA Occlusion Optimisation Algorithm

Algorithm 10 and Algorithm 11 provides the *e*PCA optimisation transformation to reduce image occlusion given the training and inference images respectively.

Algorithm 10 *e*PCA_TRAIN

- 1: **procedure** *EPCA_TRAIN_IMG*($\mathbf{I}_j (j = 1, \dots, M), \alpha, R$)
 - 2: Perform steps 2 to 8 from Algorithm 8.
 - 3: Compute \mathbf{z}_j in accordance to Eq. (5.28) and form \mathbf{Z} per Eq. (5.29).
 - 4: Compute kernel matrix \mathbf{K} per Eq. (5.30).
 - 5: Compute $\tilde{\mathbf{P}}_c$ using $[\tilde{\mathbf{P}}_c, \tilde{\Sigma}_c] = svds(\mathbf{K}, R)$.
 - 6: Set $\tilde{\Lambda}_c$ from $\tilde{\Sigma}_c$.
 - 7: Compute $\tilde{\mathbf{B}}$ from Eq. (5.34).
 - 8: Compute $\tilde{\mathbf{B}}\tilde{\mathbf{B}}^H$ and construct $\check{\mathbf{Z}}$ from Eq. (5.37) and Eq. (5.29).
 - 9: Compute pre-image $\check{\mathbf{X}}$ using Eq. (5.46).
 - 10: **return** $\check{\mathbf{X}}, \tilde{\mathbf{B}}, \tilde{\Sigma}$.
-

Algorithm 11 *e*PCA_INFERENCE

- 1: **procedure** *EPCA_INFERENCE_IMG*($\mathbf{I}_t, \alpha, \tilde{\mathbf{B}}$)
 - 2: Perform steps 2 from Algorithm 9 and form $\hat{\mathbf{x}}_t$.
 - 3: Compute \mathbf{z}_t using Eq. (5.28).
 - 4: Compute $\check{\mathbf{z}}_t$ using Eq. (5.37).
 - 5: Compute $\check{\mathbf{x}}_t$ using Eq. (5.46).
 - 6: **return** $\check{\mathbf{x}}_t$
-

5.4.4 *e*PCA Occlusion Optimisation Results

Image occlusion optimisation using *e*PCA was applied to synthetic Radarsat CAD images and flight images of the SpaceX Dragon cargo vehicle. Star patterns were added to the inference image to mimic occlusion. Figure 5.11 shows three flawed input images going through the *e*PCA process; in all three cases, the occlusion was eliminated. Ten training images of the Dragon vehicle taken by the ISS camera is provided in Fig. 5.12. All images were reduced to 320×240 resolution, and some include the SSRMS robotic manipulator during the final stage free-flyer capture event. Figure 5.13 shows the optimised and normalised training data after the *e*PCA process. Figure 5.14 shows the three Dragon

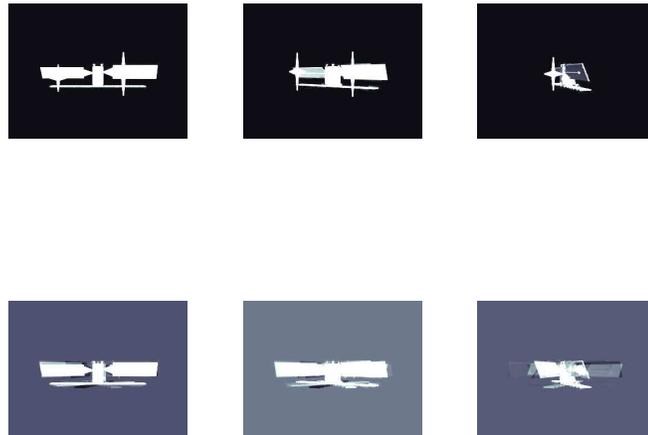


Figure 5.11: Synthetic Radarsat testing images. Top: corrupted testing images. Bottom: *ePCA* optimised output.

images with manually added occlusion. The defected image was processed by the *ePCA* and returned the optimised images underneath, where the occlusion was removed. The *ePCA* process is an elegant way to remove occlusion and improve the precision of the PCA appearance-based pose estimation matching.

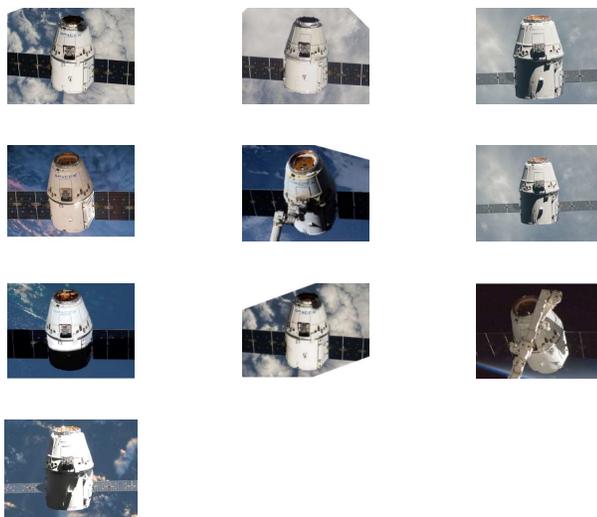


Figure 5.12: SpaceX Dragon vehicle training images.



Figure 5.13: SpaceX Dragon vehicle *ePCA* optimised training images.

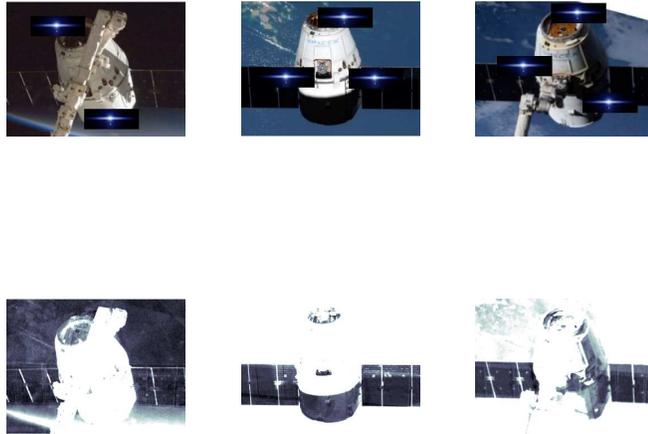


Figure 5.14: SpaceX Dragon vehicle *ePCA* optimised testing images. Top: corrupted testing images. Bottom: *ePCA* optimised output.

Chapter 6

Point-based Methods

Perspective- n -point (PnP) is a point matching approach to align stored 3D-model with the captured image by using 3D-to-2D projection. The model pose is determined by point alignment procedures that may be iterative or non-iterative. In principle, the target pose can be determined by direct point correspondence. In reality, however, image clutter and local minimum trapping from complex point correspondence energy potentials prevent robust pose determination. Practically, incremental reference estimation is computed in short time intervals which require *a priori* spacecraft state before the pose refinement process. The initial pose may come from other sensors such as LIDAR for near-range and relative GPS for far-range *etc.* . The initial target image is captured on initialisation or a previously stored target image from ground estimation may also be used. The image points are extracted using straight line inflation or image feature keypoint detection described in Chapter 2. The model points are extracted from an internal 3D-model to 2D-image projection. The model to image matching is called *Image Registration*.

This chapter describes both iterative and non-iterative PnP techniques, called the SoftPOSIT and $ePnP$ methods respectively. In our application of the SoftPOSIT approach, we align target image corner points and 3D model vertices. The alignment is performed assuming weak perspective, and uses a so-called soft-assign iteration. The SoftPOSIT iteration success is dependent on the quality of the iteration parameter initialisation. We introduce two novel methods for finding the SoftPOSIT initialisation parameter and improve the SoftPOSIT solution by avoiding local minimum traps. We also introduce a new pose estimation process using PnP solvers using image homography and model projection. Our approach works with internal models with low number of vertices since we do not use the actual image when performing the PnP operation. The SoftPOSIT and $ePnP$ software is provided in Appendix C.6 and C.7 respectively.

6.1 PnP Method Overview

Point-based pose estimation has three main steps: generating image points from the internal model, extracting image points from the captured image, and performing PnP matching. Figure 6.1 provides the full point-based pose estimation process. The internal model is generated by assembling basic geometric elements, model points are either selected from corner vertices or by adjacent line point inflation. If point image features are used, then an initial template image is captured. The template image is a calibration point; we assume the pose of the template is available from an external source. The image features extracted from this template will be used for pose matching. Also, as part of the initialisation, the camera intrinsic properties are retrieved from a storage file or recomputed using standard camera calibration techniques described in Chapter 2, Sec. 2.3.2.

Next, the target CS image is captured by using a synthetic or real-world camera; this image is enhanced using techniques discussed in Chapter 2 Sec. 2.1. If the image points are corners, then corners points are extracted from the edge gradient image as described in Chapter 2 Sec. 2.2. Alternatively, point features including their descriptors are extracted as described in Chapter 2 Sec. 2.3. The feature points are matched with the template image, and RANSAC is used to increase point matching robustness. We also develop a pose limit sentinel to prevent large motions due to outlier matching. A popular PnP approach ^a is to directly use the feature points from the image to match with feature points estimated from the model plane and light ray intersections [417,418]. This approach requires manual initialisation defining the template pose with respect to the template features. Our approach uses image homography to compute the transformation from the template to the current image; then we transform the 3D model template projection to the new image using the computed homography matrix.

Finally, we use iterative or non-iterative PnP to match the internal 3D model projection with the homography transformed template points and apply the necessary kinematic transformations to compute the position and orientation of the target CS in the required frame of reference. Our approach allows an exact match of the image and model points. The 3D model itself can be a simplified shape and fast to compute. The following sections describe each point-based pose estimation steps and related enhancements in detail.

^ahttps://docs.opencv.org/trunk/dc/d2c/tutorial_real_time_pose.html

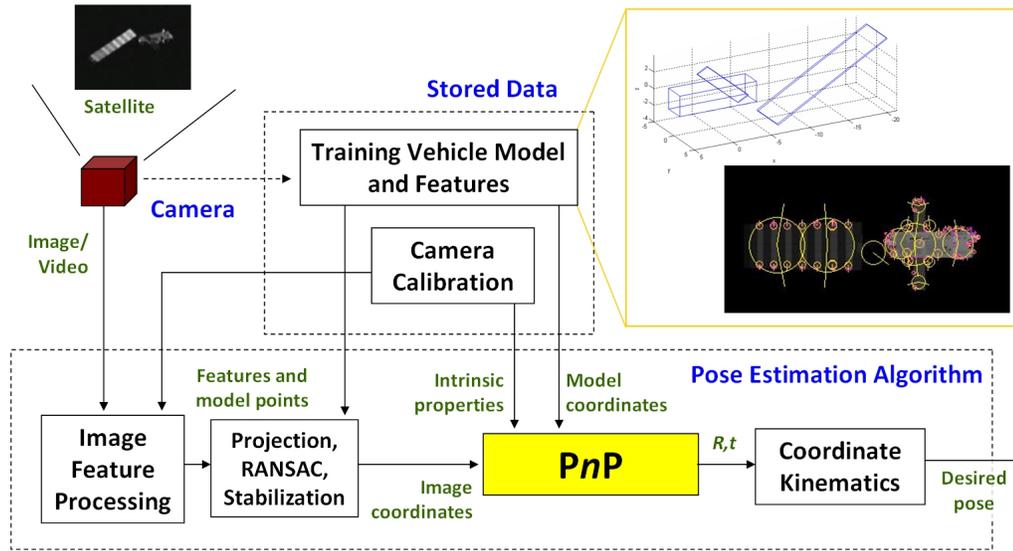


Figure 6.1: Point-based pose estimation pipeline overview.

6.2 Internal 3D Model Generation

The target CS internal 3D model is generated by their corner vertices. Basic geometric shapes such as rectangle, cone, ellipse and cylinder can be used to build a more geometrically complex spacecraft. The basic shapes are provided in Fig. 6.2. To increase robustness of the PnP solution, we apply point inflation on the internal 3D model, details of point inflation is discussed in Chapter 2 Sec. 2.2.2. Points are added on the adjacent lines from the *adjacent matrix*. In the CubeSat example, 10 points were created along each adjacent line. Figure 6.3 provides the resulting internal model after point inflation. The algorithm and software code for geometry construction and point inflation are provided in Appendix C.2.

6.3 Camera Model

Let us define a Servicing Spacecraft (SS) body frame, \mathcal{F}_{SB} , equipped with a single camera positioned at the camera frame, \mathcal{F}_{VW} , pointed towards a Client Satellite (CS) body frame, \mathcal{F}_{CB} . The frames \mathcal{F}_{SB} and \mathcal{F}_{CB} are located at the spacecraft Centre of Mass (CoM). The camera frame \mathcal{F}_{VW} has its Z axis pointed outwards from the boresight of the camera, with

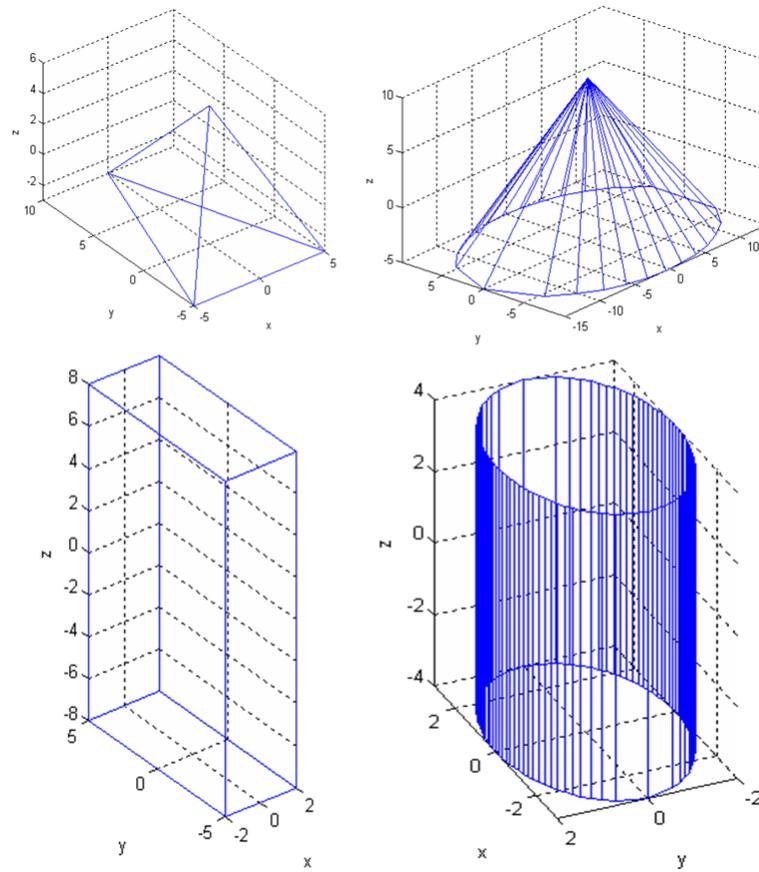


Figure 6.2: Elemental building shapes.

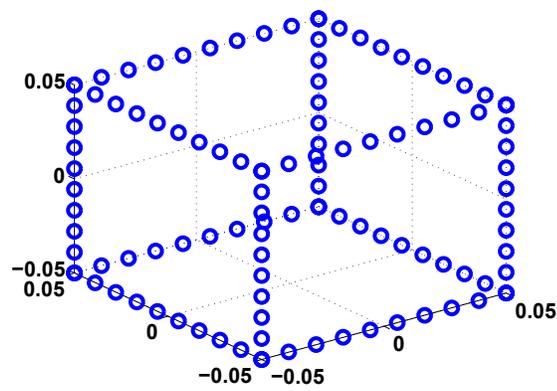


Figure 6.3: Internal model adjacent line point inflation

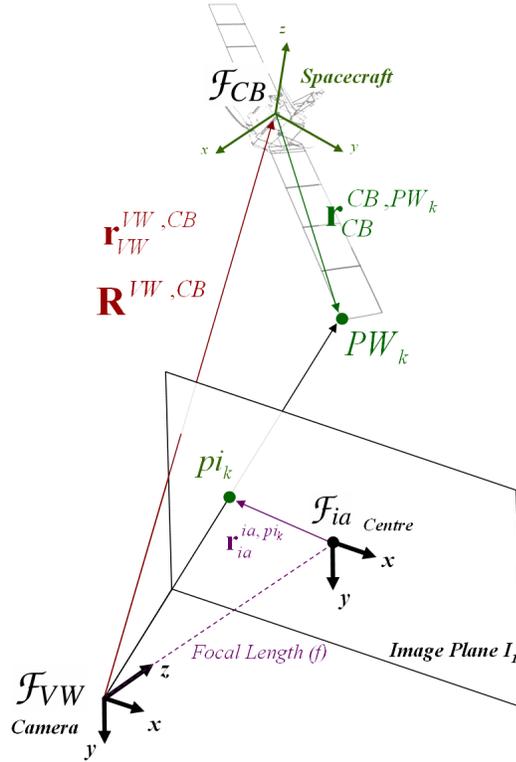


Figure 6.4: Camera coordinate system definitions.

the Y axis pointed vertically downwards and X completing the right-hand. The frame \mathcal{F}_{ia} is centred on the camera image with its X and Y axes being parallel to \mathcal{F}_{VW} . Figure 6.4 provides the various coordinate systems. Let us define PW_k be the k^{th} point on the target that is projected on the image plane. The total number of points from the target model is M , where $k \in 1 \dots M$; these are the stored points of the target body in the 3D model. We define the vector to these 3D model points relative to and expressed in the body frame as $\mathbf{P}_k^w = \mathbf{r}_{CB, PW_k}^{CB, PW_k}$. The objective is to determine the pose, *i.e.* \mathbf{t} and \mathbf{R} of the target with respect to the camera defined as follows,

$$\mathbf{t} = \mathbf{r}_{VW}^{VW, CB} = \begin{bmatrix} t_x & t_y & t_z \end{bmatrix}^T, \quad (6.1)$$

$$\mathbf{R} = \mathbf{R}^{VW, CB} = \begin{bmatrix} \mathbf{R}_x^T \\ \mathbf{R}_y^T \\ \mathbf{R}_z^T \end{bmatrix}, \quad (6.2)$$

where \mathbf{t} is the position of \mathcal{F}_{CB} with respect to \mathcal{F}_{VW} expressed in \mathcal{F}_{VW} and \mathbf{R} is the rotation matrix rotating \mathcal{F}_{CB} to \mathcal{F}_{VW} , and \mathbf{R}_x^T , \mathbf{R}_y^T , and \mathbf{R}_z^T are row matrices of \mathbf{R} . These are also the \mathcal{F}_{VW} unit vectors expressed in \mathcal{F}_{CB} . The parameters \mathbf{t} and \mathbf{R} is sometimes referred to as the camera extrinsic properties. We define the model points observed and expressed by the camera frame as $\mathbf{P}_k^c = \mathbf{r}_{VW}^{VW, PW_k}$, this vector is related to the model vector and the camera extrinsic properties as,

$$\mathbf{r}_{VW}^{VW, PW_k} = \mathbf{r}_{VW}^{VW, CB} + \mathbf{R}^{VW, CB} \mathbf{r}_{CB}^{CB, PW_k}. \quad (6.3)$$

The projection of PW_k on the image frame \mathcal{F}_{ia} is denoted as pi_k , and $\mathbf{r}_{VW}^{VW, PW_k}$'s image projection is

$$\mathbf{r}_{ia}^{ia, pi_k} = \frac{f}{Z_k^c} \mathbf{\Upsilon} \mathbf{r}_{VW}^{VW, PW_k}, \quad (6.4)$$

where $\mathbf{\Upsilon} = \begin{bmatrix} \mathbf{1} & \mathbf{0} \end{bmatrix}$, and $\mathbf{1}$ is an 2×2 identity matrix, and $\mathbf{0}$ is a 2×1 zero matrix, and Z_k^c is the Z component of r_{VW}^{VW, PW_k} .

Let us define N as the number of detected image points, where $j \in 1 \dots N$, and pi_j denoting the j^{th} image point. The position of the image point relative to the image centre is $\mathbf{r}_{ia}^{ia, pi_j} = \begin{bmatrix} x_j & y_j \end{bmatrix}^T$. The image pixel origin is define in the upper left corner of the image. The pixel coordinate is computed as the linear transformation $\mathbf{p}_j = \mathbf{r}_{ia}^{ia, pi_j} + \begin{bmatrix} \tilde{o}_x & \tilde{o}_y \end{bmatrix}^T$, where \tilde{o}_i , $i \in x, y$, is the pixel distance from the image origin to the image centre. The image and model points in homogeneous coordinates are defined as $\tilde{\mathbf{p}}_j = \begin{bmatrix} \mathbf{p}_j^T & 1 \end{bmatrix}^T = \begin{bmatrix} u_j & v_j & 1 \end{bmatrix}^T$ and $\tilde{\mathbf{P}}_k^w = \begin{bmatrix} \mathbf{P}_k^{wT} & 1 \end{bmatrix}^T$ respectively. The camera intrinsic properties are grouped into the following matrix,

$$\mathbf{K} = \begin{bmatrix} fS_x & 0 & \tilde{o}_x \\ 0 & fS_y & \tilde{o}_y \\ 0 & 0 & 1 \end{bmatrix}, \quad (6.5)$$

where, f is the focal length, S is the pixel scale conversion. Finally, the camera model expressed in an alternative form from Eq. (6.4) is,

$$Z_k^c \tilde{\mathbf{p}}_k = \mathbf{K} \begin{bmatrix} \mathbf{R} & \mathbf{t} \end{bmatrix} \tilde{\mathbf{P}}_k^w, \quad (6.6)$$

where Z_k^c is the Z component of \mathbf{P}_k^c .

6.4 Feature Matching

Point-based pose estimation may extract points from corners, edge lines, and blob keypoints. Direct PnP matching of corners or points inflated edge lines do not require feature matching. Alternatively, the image features are extracted from the processed template image and the input image. Image processing may be normalisation, HE or CLAHE; the fastest approach that will work for most images is normalisation. A detailed discussion of generating image feature keypoints and descriptors is in Chapter 2 Sec. 2.3. Our method computes the homography transformation between the input image and the template through feature matching. Descriptor feature matching is performed using the k -Nearest Neighbour (kNN) method. Specifically, two nearest neighbours L_2 -norm distances are compared to eliminate incorrect matches. Lowe [15] suggest the true match should have a small first to second match distance ratio since there can only be one distinctive match to the template image feature. The kNN method can also be collapsed into one-dimension (1D) by comparing the angle between two descriptor vectors. In the polar domain, angles of the first and second match are compared instead of the L_2 -norm distance. We use 0.6 as the match distance ratio based on our tests. In our experience, the 1D angle approach is faster, however, it is less precise than the L_2 -norm distance matching.

In the following example, we demonstrate the feature matching method using the Envisat model, SIFT [15], and Shi-Tomasi [325] corners with the BRIEF image descriptor. The feature matching image is provided in Fig. 6.5, where 15 degree rotation between the template and the input is compared. The feature matching results are provided in Fig. 6.6, where estimated matches are compared with good matches. An estimated match is the best match suggested by the kNN algorithm. A true match is the ground truth match. A good match is defined as an estimated match that is also a true match. The BRIEF feature has zero real matches after 40 degrees yaw angle rotation although kNN matching falsely believe 20 matches still exist; this is because the BRIEF descriptor is based on corner keypoints that do not have orientation signature and is therefore not rotationally invariant. On the other hand, SIFT computes keypoint orientation and can match keypoints after a large 135 degrees rotational change. Before 15 degrees, the SIFT and BRIEF match performance are

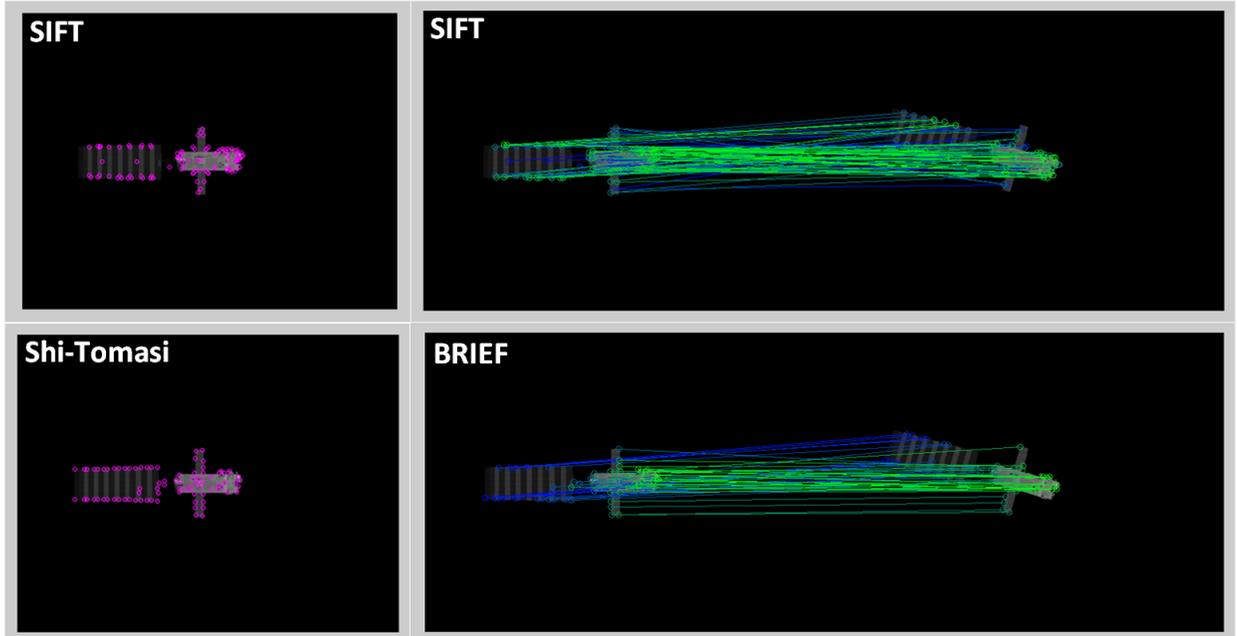


Figure 6.5: Envisat feature matching, top image is SIFT keypoints and matching, the bottom image is Shi-Tomasi corner keypoints and matching.

comparable, suggesting the BRIEF feature is useful for sequential frame matching where inter-frame angular variation is small. The match quality is the percentage good matches overall matches, and it indicates the number of outliers. Figure. 6.7 shows BRIEF percentage good match drops to zero with increasing yaw angle, and for low rotation angles, the match quality is relatively consistent for match ratios above 0.6. Future experiments may explore faster matching methods such as FLANN [350], Best Bin First [15], and Energy-based multi-model Fitting-and-Matching [419].

6.5 Homography, RANSAC and Sentinel

The homography matrix transforms a pixel coordinate from one image to another, it encodes the scale, rotation, and affine transformations of the target image. Hereon forward we designate the template image as the training image, and the input image as the query image. The homography method [404] computes the transformation between the training and the query image using the following,

$$\tilde{\mathbf{p}}_q = \mathbf{H}\tilde{\mathbf{p}}_t, \quad (6.7)$$

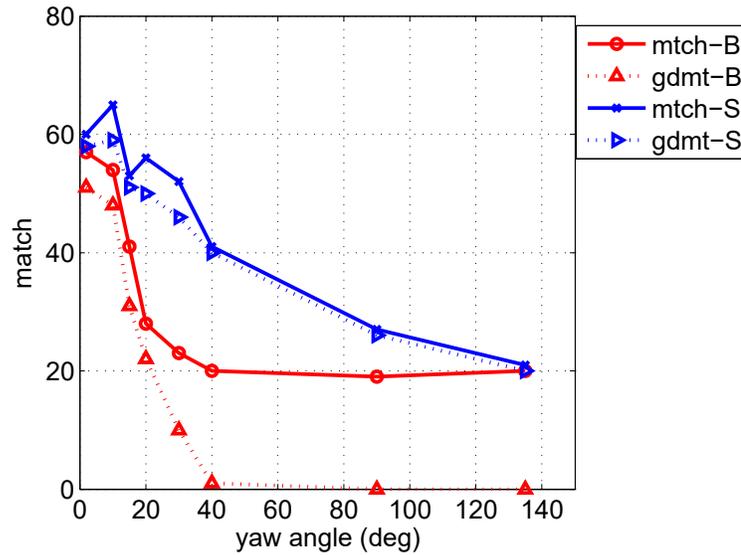


Figure 6.6: SIFT vs. BRIEF matches, where *mtch* is estimated matches, *gdmt* is good matches, *B* is BRIEF, and *S* is SIFT.

where \mathbf{H} is the homography matrix, and $\tilde{\mathbf{p}}$ is the homogeneous pixel coordinate for the training, t , and the query, q , images. The homography matrix is not guaranteed in general, other robustness measures such as a movement limit sentinel were developed and will be discussed later.

Wrong feature matches will cause large errors in the homography matrix. We may limit this uncertainty by applying a voting technique called RANdom Sample and Consensus (RANSC) [94] to remove outliers. Our RANSAC algorithm randomly selects s sample points from the entire query set and compute the homography matrix; then we apply the homography matrix to the training points using Eq. (6.7). We compare the query points with the transformed training points and discard any outliers outside a tolerance of 3 pixels as shown in Fig. 6.8. We compute the sum of all RSS distance differences between the remaining query and transformed training points over T trials. The optimal number of RANSAC trials is computed using the following equation,

$$T = \frac{\log(1-p)}{\log(1-(1-e)^s)}, \quad (6.8)$$

where p is the desired success probability, and e is the outlier ratio. Testing show using 0.6 matching ratio and 20 sample points to achieve 90% success probability requires 5 trials

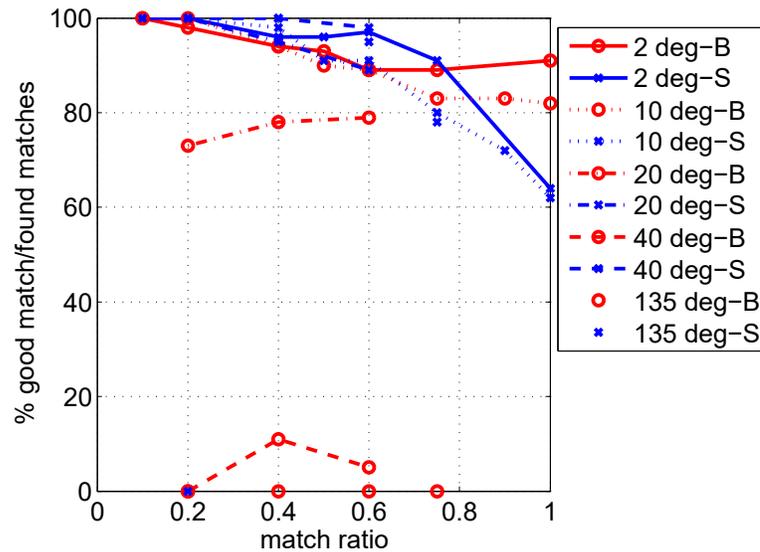


Figure 6.7: Match quality relative to match ratio, B is BRIEF, S is SIFT.

and 2 ms computation time.

We use a movement limit sentinel to handle invalid homography transformations and avoid passing wrongly projected model points to the P_nP solver. We develop a movement limit sentinel to evaluate the model's maximum movement points. We assume the target and the tracker to have realistic velocities that will not exceed some allowable pixel range in one frame. Given the short time between frames, the pose is held constant for the next frame. The movement range adaptively increases as more erroneous frames are returned. Upon receiving a good frame, the movement range is restored to the original value. The RANSAC and movement limit sentinel software are in Appendix C.7.

6.6 Model Projection P_nP

Our point-based pose estimation is unique because we do not directly match the model points with points extracted from the raw image. We find the raw image points to have large discrepancies with the internal 3D model projected points. The error is amplified when the number of sample points is low. If large amounts of sample points are used, then the P_nP runtime will increase. To balance maximum sample points and low run time, we transform the internal 3D model to 2D projection points from the template image to the query image using the homography matrix. We then use the P_nP solver to match the 3D

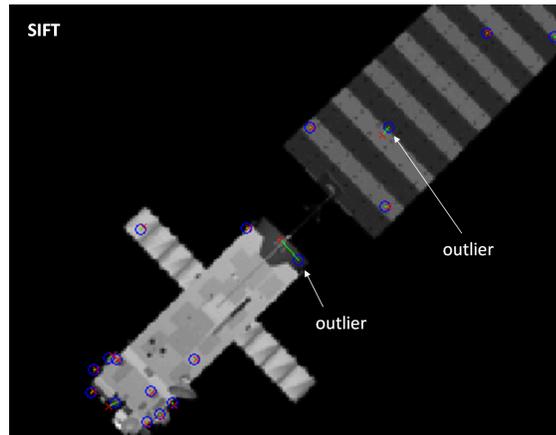


Figure 6.8: SIFT feature matching outliers.

model projection to the homography transformed points to compute the 6-DOF pose; this ensures a one-to-one match of all homography transformed points and points projected by the internal model. Consequently, we can use a small internal 3D model of a simplified spacecraft or some elementary shape. Future developments could extend this method into semi-supervised or unsupervised problems.

6.7 SoftPOSIT

David *et al.* [100] solves the PnP problem by using Simulated Annealing (SA) [264] and Scaled Orthographic Projection (SOP) [262]; this is the so-called SoftPOSIT method; it is a combination of *Softassign* [263, 420] and *Pose from Orthography and Scaling with Iterations* (POSIT). SoftPOSIT is an iterative point correspondence scheme minimising a global energy function based on model and image point differences; it was used in thermal image ground vehicle pose estimation [421], and spacecraft pose estimation [422]. SoftPOSIT can produce fast pose estimation for small pose convergence; however, for large initial misalignments, this algorithm can return erroneous results due to local minimum trapping and ill-defined initialisation parameters. We provide novel enhancements to these shortcomings in the next section.

6.7.1 SoftPOSIT Formulation

Let us define the *scaling ratio* s as f/t_z and the *prospective ratio* w_k as Z^c/t_z , such that the prospective scaling term f/Z^c can be replaced with s/w_k . In the weak prospective view, where the target object is far away from the camera and the FOV is not abnormally large, then w_k is nearly one. The w_k term can be written as

$$w_k = \frac{\mathbf{R}_z^T \mathbf{P}_k^w}{t_z} + 1. \quad (6.9)$$

The pose matrices are defined as $\mathbf{Q}_x \triangleq s \begin{bmatrix} \mathbf{R}_x^T & t_x \end{bmatrix}^T$, $\mathbf{Q}_y \triangleq s \begin{bmatrix} \mathbf{R}_y^T & t_y \end{bmatrix}^T$. The distance between the projected model points and the camera points is

$$d_{jk}^2 = (\mathbf{Q}_x^T \tilde{\mathbf{P}}_k^w - w_k u_j)^2 + (\mathbf{Q}_y^T \tilde{\mathbf{P}}_k^w - w_k v_j)^2. \quad (6.10)$$

A *Global Objective Function* is formulated as

$$E = \sum_{j=1}^N \sum_{k=1}^M m_{jk} (d_{jk}^2 - \alpha), \quad (6.11)$$

where m_{jk} and α are the weights and control parameter respectively that will be defined in detail later. For a maximum correspondence between image and model points, the partial derivative of E with respect to the pose matrices is set to zero. Let us define,

$$\mathbf{L} = \sum_{j=1}^N \sum_{k=1}^M m_{jk} \tilde{\mathbf{P}}_k^w \left(\tilde{\mathbf{P}}_k^w \right)^T, \quad (6.12)$$

then the maximum correspondence partial derivative of E will result in the following pose vectors,

$$\begin{aligned} \mathbf{Q}_x &= \mathbf{L}^{-1} \sum_{j=1}^N \sum_{k=1}^M m_{jk} w_k x_j \tilde{\mathbf{P}}_k^w \\ \mathbf{Q}_y &= \mathbf{L}^{-1} \sum_{j=1}^N \sum_{k=1}^M m_{jk} w_k y_j \tilde{\mathbf{P}}_k^w. \end{aligned} \quad (6.13)$$

The weights m_{jk} are computed at every step based using SA as follows,

$$m_{jk} = \gamma \exp(-\beta(d_{jk}^2 - \alpha)), \quad (6.14)$$

where γ is a normalisation factor and α allows amplification of the d_{jk} distance [100], and β is the iteration control parameter to be discussed in detail in Sec. 6.8.2. The corresponding distance between image and model points can be grouped into a matrix as follows,

$$\mathbf{D} = \begin{bmatrix} d_{1,1}^2 & d_{1,2}^2 & \cdots & d_{1,M}^2 \\ d_{2,1}^2 & d_{2,2}^2 & \cdots & d_{2,M}^2 \\ \vdots & \vdots & & \vdots \\ d_{N,1}^2 & d_{N,2}^2 & \cdots & d_{N,M}^2 \end{bmatrix}. \quad (6.15)$$

Once the pose matrices are computed, x and y rows of the rotation matrix axis can be found using SVD,

$$\begin{bmatrix} \mathbf{R}_x & \mathbf{R}_y \\ t_z & t_z \end{bmatrix} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T. \quad (6.16)$$

The rotation axis and t_z can be computed by

$$\begin{bmatrix} \mathbf{R}_x & \mathbf{R}_y \end{bmatrix} = \mathbf{U}\mathbf{\Upsilon}^T\mathbf{V}^T, \quad (6.17)$$

$$t_z = \frac{2}{\Sigma_{1,1} + \Sigma_{2,2}}, \quad (6.18)$$

where $\Sigma_{i,i}$, $i \in 1, 2$, is the first two diagonal values from the $\mathbf{\Sigma}$ matrix of Eq. (6.16), and $\mathbf{\Upsilon}$ is the same identity and zero matrix from Eq. (6.4). Finally, \mathbf{R}_z can be computed from the cross product of \mathbf{R}_x and \mathbf{R}_y . SoftPOSIT iteration is terminated if the estimated pose converges below a user-defined tolerance.

6.8 Enhanced SoftPOSIT

The SoftPOSIT approach takes advantage of the weak prospective property, and for small misalignments between the initial internal 3D model pose projection and the image points, it can quickly converge to a correct pose solution; however, when the initial misalignment

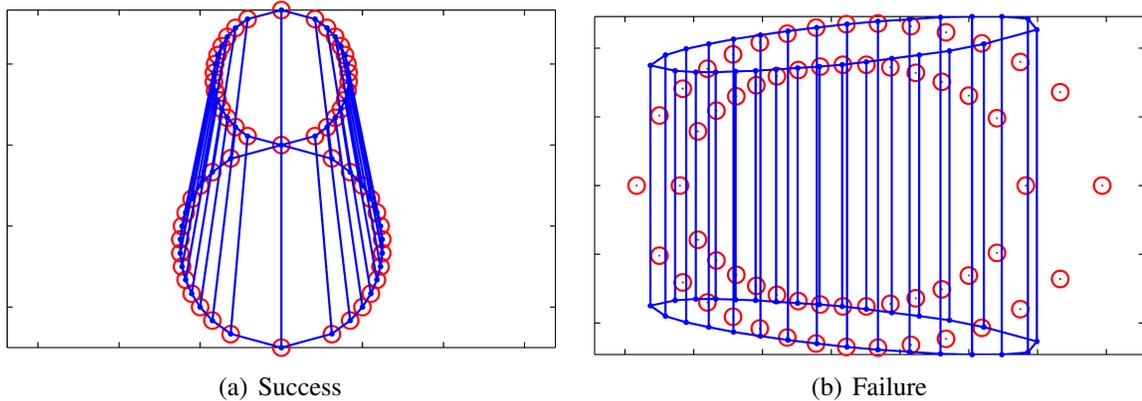


Figure 6.9: 30-point cylinder SoftPOSIT pose estimation; the red circles are image points, the blue points are final estimated model points.

is large, the iteration process can often converge to a false solution as a result of local minimum trapping. An example of the local minimum trapping is demonstrated in Fig. 6.13, where the pose estimation of a 30-points cylinder is computed. Figure 6.13(a) shows the SoftPOSIT method produces the correct pose estimation in one cylinder configuration; whereas in Fig. 6.13(b) shows with the same initial condition (IC) pose, but with a different cylinder image, the final pose converges to the wrong pose. The wrong pose solution, however, is a local minimum energy solution. In the following sections, we introduce two novel initialisation techniques to minimise the chances of local minimum trapping. These methods include performing a global minimum search and adaptive initialisations of the iteration control parameter. We perform exhaustive testing of these initialisation methods using basic element shapes and present the performance results.

6.8.1 Global Minimum Search

Local minimum trapping is a major cause in the SoftPOSIT pose estimation failure [100, 421, 422]; it is caused by a complex potential formation in the object pose hyperspace. Under large initial misalignment conditions, it is easy for the iteration process to converge into local minimum correspondence energy and cannot exit. The softPOSIT iteration process is rooted in SA [263] and optimised by the Sinkhorn's normalisation process [420]. Simulated annealing [264] models after the metal annealing formation process where the metal crystal lattice solidifies after some exponentially decaying heating period. The SoftPOSIT

iteration loses the ability to exit the current local minimum exponentially. Our approach is to initially force the model pose into multiple directions and based on the misalignment performance to decide on the most likely path.

We make two important observations in the convergence process. First, because of the exponential characteristics as shown in Eq. (6.14), a majority of pose error is removed during the first several steps of the annealing process. Therefore, one may try different initial orientations for a few steps before selecting the final path direction. Secondly, in most cases, after the same number of iteration steps, the global minimum solution usually has lower overall correspondence than local minimum solutions. Based on the two observations, we propose a pose IC switching logic to search for the global minimum solution direction. Let n be the number of pose directions we will use to check for the global minimum solution. We use the axis angle ϕ about axis \vec{a}_i , where $i \in 1 \dots n$. The angle ϕ should be large enough to bring the temporary solution away from the local minimum. For example, we set ϕ to 90 degrees, and select $n = 4$ axis directions, $(1, 0, 0)$, $(0, 1, 0)$, $(0, 0, 1)$, and $(1, 1, 1)$, with respect to the initial orientation.

We summarise the second observation as *the smallest maximum of the closest image to model correspondence for all initial orientations is the likely path towards the global minimum*. We may formally establish the above statement by writing the correspondence matrix from Eq. (6.15) as an array of column matrices $\mathbf{D} = \begin{bmatrix} \mathbf{D}_1 & \dots & \mathbf{D}_M \end{bmatrix}$, these columns represent the correspondence from all the detected image points to one model projected point. The minimum value in the k^{th} column represent the closest image to model correspondence for the k^{th} model point, that is,

$$\hat{D}_{ki} = \operatorname{argmin}_{j \in N} \{D_{jki}\}. \quad (6.19)$$

The maximum of the filtered distance represent the *goodness* resulting from optimising from the i^{th} initial orientation,

$$\check{D}_i = \operatorname{argmax}_{k \in M} \{\hat{D}_{ki}\}. \quad (6.20)$$

Finally, the minimum \check{D}_i signals the most likely path to the global minimum solution,

$$\hat{D}_g \triangleq \operatorname{argmin}_{i \in n} \{\check{D}_i\}. \quad (6.21)$$

In the spirit of SA optimisation, we name this pose initialisation phase as *preheating*.

6.8.2 Control Parameter based on Correspondence

In Eq. (6.14), the iteration control parameter β is used to control the annealing process. When β is low, the weighting will be high and vice-versa. The control parameter β is mostly estimated empirically, David *et al.* [100] estimated 0.0004 while Gold *et al.* [263] used 0.00091. Poorly selected β and correspondence matrix, \mathbf{D} , combinations will lead to unrealistic outputs from the exponential term in Eq. (6.14). We propose an adaptive β initialisation based on the correspondence matrix as,

$$\beta_0 = F \frac{(M + N)/2}{\operatorname{tr}(\mathbf{D})}, \quad (6.22)$$

where F is some scaling constant tested to be 2, and $\operatorname{tr}(\mathbf{D})$ is the trace of the \mathbf{D} matrix. Since the trace of a matrix is also the sum of its eigenvalues, Eq. (6.22) have better numerical compatibility between β_0 and the correspondence distances.

Eq. (6.22) provides an iteration control parameter effective mean in case of an optimisation restart. The causes for the restart can be because an ill-conditioned and non-invertible \mathbf{L} matrix; or pose divergence during the correspondence minimisation. Under these conditions, β_0 is recomputed by Eq. (6.22).

6.8.3 Control Parameter based on Centroid Matching

The SA [264] iteration control parameter in Eq. (6.14) is analogous to the inverse of the annealing temperature [423]. The exponential in the ‘cooling’ period requires an appropriate initial ‘temperature’ selection such that the weighting, m_{jk} , do not become zero or infinity. We propose a *centroid matching* approach to generate the SA iteration control parameter. The β_0 parameter is computed based on the image condition and latest point

correspondence, and it prevents numerical anomalies in the weighting generation. We initially estimate the target CS CoM from the image point centroid. The image centroid is formulated as,

$$\bar{\mathbf{r}}_{ia}^{ia,pi} = \sum_{j=1}^N \frac{r_{ia}^{ia,pij}}{N}, \quad (6.23)$$

Combining Eqs. (6.4) and (6.23),

$$\frac{\bar{\mathbf{r}}_{ia}^{ia,pi}}{f} = \Upsilon \frac{\bar{\mathbf{r}}_{VW}^{VW,PW}}{t_z}. \quad (6.24)$$

We approximate the image centroid as the CS CoM

$$\bar{\mathbf{r}}_{VW}^{VW,PW} \triangleq \mathbf{r}_{VW}^{VW,CB}. \quad (6.25)$$

Substituting Eqs. (6.1) and (6.24) into Eq. (6.25) will produce the x and y centroid position as a function of projected CoM.

$$\frac{\bar{\mathbf{r}}_{ia}^{ia,pi}}{f} \triangleq \begin{bmatrix} t_x/t_z \\ t_y/t_z \end{bmatrix}. \quad (6.26)$$

Let Γ be the inverse of the matrix \mathbf{L} , it can be written as,

$$\begin{aligned} \Gamma &= \mathbf{L}^{-1} \\ &= \begin{bmatrix} \Gamma_{13} \\ \Gamma_4 \end{bmatrix}, \end{aligned} \quad (6.27)$$

where Γ_{13} is the first three rows of the inverse \mathbf{L} matrix and Γ_4 is the fourth row of the inverse \mathbf{L} matrix. Combine Eqs. (6.13) and (6.27),

$$\begin{aligned} \begin{bmatrix} \mathbf{Q}_x/f \\ \mathbf{Q}_y/f \end{bmatrix} &= \begin{bmatrix} \mathbf{R}_x^T/t_z \\ t_x/t_z \\ \mathbf{R}_y^T/t_z \\ t_y/t_z \end{bmatrix} \\ &= \begin{bmatrix} \left[\begin{array}{c} \Gamma_{13} \\ \Gamma_4 \end{array} \right] \sum_{j=1}^N \sum_{k=1}^M m_{jk} w_k \frac{x_j}{f} \widehat{\mathbf{P}}_k \\ \left[\begin{array}{c} \Gamma_{13} \\ \Gamma_4 \end{array} \right] \sum_{j=1}^N \sum_{k=1}^M m_{jk} w_k \frac{y_j}{f} \widehat{\mathbf{P}}_k \end{bmatrix}. \end{aligned} \quad (6.28)$$

Rewriting the equations in matrix format, we let

$$\begin{bmatrix} \mathbf{U}/f \\ \mathbf{V}/f \end{bmatrix} = \begin{bmatrix} \frac{x_1}{f} & \dots & \frac{x_N}{f} \\ \frac{y_1}{f} & \dots & \frac{y_N}{f} \end{bmatrix}, \quad (6.29)$$

$$\mathbf{M} = \begin{bmatrix} m_{11} & \dots & m_{1M} \\ \vdots & \ddots & \vdots \\ m_{N1} & \dots & m_{NM} \end{bmatrix}, \quad (6.30)$$

$$\begin{aligned} \mathbf{P}_M^T &= \begin{bmatrix} \tilde{\mathbf{P}}_1^w & \dots & \tilde{\mathbf{P}}_M^w \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{r}_{CB}^{CB,PW_1} & \dots & \mathbf{r}_{CB}^{CB,PW_M} \\ 1 & \dots & 1 \end{bmatrix}, \end{aligned} \quad (6.31)$$

w_k for all the image points can be grouped into a matrix, by combining Eq. (6.31) and Eq. (6.9),

$$\begin{aligned} \mathbf{w}_M &= \begin{bmatrix} w_1 \\ \vdots \\ w_M \end{bmatrix} \\ &= \mathbf{P}_M \begin{bmatrix} \mathbf{R}_z/t_z \\ 1 \end{bmatrix}. \end{aligned} \quad (6.32)$$

Also, let

$$h_k = w_k \mathbf{\Gamma}_4 \tilde{\mathbf{P}}_k^w, \quad (6.33)$$

$$\begin{aligned} \mathbf{h}_M &= \begin{bmatrix} h_1 \\ \vdots \\ h_M \end{bmatrix} \\ &= \begin{bmatrix} w_1 \\ \vdots \\ w_M \end{bmatrix} \circ \begin{bmatrix} \mathbf{\Gamma}_4 \tilde{\mathbf{P}}_1^w \\ \vdots \\ \mathbf{\Gamma}_4 \tilde{\mathbf{P}}_M^w \end{bmatrix} \\ &= \mathbf{w}_m \circ \left(\mathbf{P}_M \mathbf{\Gamma}_4^T \right), \end{aligned} \quad (6.34)$$

where \circ is the Hadamard point-wise multiplication operator. Extract T_x and T_y from Eq. (6.28) and substitute in Eq. (6.33), and rewrite as,

$$\begin{aligned} \begin{bmatrix} t_x/t_z \\ t_y/t_z \end{bmatrix} &= \begin{bmatrix} \sum_{j=1}^N \frac{x_j}{f} \sum_{k=1}^M m_{jk} h_k \\ \sum_{j=1}^N \frac{y_j}{f} \sum_{k=1}^M m_{jk} h_k \end{bmatrix} \\ &= \begin{bmatrix} \frac{x_1}{f} & \cdots & \frac{x_N}{f} \\ \frac{y_1}{f} & \cdots & \frac{y_N}{f} \end{bmatrix} \begin{bmatrix} \sum_{k=1}^M m_{1k} h_k \\ \vdots \\ \sum_{k=1}^M m_{Nk} h_k \end{bmatrix} \\ &= \begin{bmatrix} \frac{x_1}{f} & \cdots & \frac{x_N}{f} \\ \frac{y_1}{f} & \cdots & \frac{y_N}{f} \end{bmatrix} \begin{bmatrix} m_{11} & \cdots & m_{1M} \\ \vdots & \ddots & \vdots \\ m_{N1} & \cdots & m_{NM} \end{bmatrix} \begin{bmatrix} h_1 \\ \vdots \\ h_M \end{bmatrix}. \end{aligned} \quad (6.35)$$

Substitute Eq. (6.29), Eq. (6.30), and Eq. (6.34) into Eq. (6.35),

$$\begin{bmatrix} t_x/t_z \\ t_y/t_z \end{bmatrix} = \begin{bmatrix} \mathbf{U}/f \\ \mathbf{V}/f \end{bmatrix} \mathbf{M} \mathbf{h}_M. \quad (6.36)$$

The weighting matrix is defined by Eq. (6.14), let

$$\mathbf{D}_3 = \begin{bmatrix} \mathbf{U}/f \\ \mathbf{V}/f \end{bmatrix} \gamma \exp(-\beta_0(\mathbf{D} - \alpha \mathbf{1})). \quad (6.37)$$

Combine initial assumption for centroid Eq. (6.26), Eq. (6.37) becomes

$$\frac{\bar{\mathbf{r}}_{ia}^{ia,pi}}{f} \triangleq \mathbf{D}_3 \mathbf{h}_M, \quad (6.38)$$

write $\bar{\mathbf{r}}_{ia}^{ia,pi}/f$ as \mathbf{r} and square the left hand side, Eq. (6.39) becomes

$$\mathbf{r}^T \mathbf{r} \triangleq \mathbf{h}_M^T \mathbf{D}_3^T \mathbf{D}_3 \mathbf{h}_M. \quad (6.39)$$

Rewrite $\mathbf{r}^T \mathbf{r}$ as r_2 , and $\mathbf{D}_3^T \mathbf{D}_3$ as \mathbf{D}_4 Eq. (6.39) becomes

$$r_2 \triangleq \mathbf{h}_M^T \mathbf{D}_4 \mathbf{h}_M. \quad (6.40)$$

To solve Eq. (6.40), let

$$\mathfrak{F} \triangleq \mathbf{h}_M^T \mathbf{D}_4 \mathbf{h}_M - r_2 = 0. \quad (6.41)$$

The rate of change of \mathfrak{F} with respect to β is,

$$\mathfrak{F}' = \frac{d\mathfrak{F}}{d\beta}. \quad (6.42)$$

In finite steps, Eq. (6.43), becomes

$$\mathfrak{F}'_{n-1} = \frac{\mathfrak{F}_n - \mathfrak{F}_{n-1}}{\beta_n - \beta_{n-1}}, \quad (6.43)$$

using Newton's method of gradient descent, Eq. (6.41) becomes

$$\beta_{n+1} = \beta_n - \frac{\mathfrak{F}_n}{\mathfrak{F}'_{n-1}}. \quad (6.44)$$

Our experiment show β_0 can converge to 10^{-14} error within 30 iterations. For cases where gradient descent cannot produce the root value, we use Eq.(6.22) to compute the initial β_0 .

Table 6.1: Descriptions of the model shapes, number of points per model, and β_0 initial values.

Item	Item Description	Dimension	β_0
1	Cube	1x1x1	1e-4
2	Rectangle	1x2x3	1e-2
3	5-pt Cylinder	2x2x2	1e-4
4	15-pt Cylinder	2x2x2	1e-4
5	30-pt Cylinder	2x2x2	1e-4

6.8.4 Enhanced Performance

We perform 3500 simulations to test the SoftPOSIT enhancement method; the total number of runs includes 875 cases per batch over four batches. The four simulation batches are the baseline SoftPOSIT [100], centroid matching, and global minimum search without and with centroid matching. The 875 cases include a combination of object shape and number of points, initial model pose, and target object pose. We use basic elementary shapes with a different number of points as provided in Table 6.1. The orientations are given in Euler Pitch-Yaw-Roll (PYR) rotational sequences. The boundaries of the pose misalignment hyperspace is 10 length unit in X and Y , 30 length unit in Z , and 225 degrees in orientation. The 3D model IC pose is provided in Table 6.2. The misalignments are relative to the camera coordinate system, \mathcal{F}_{VW} . The Z direction is positive since the 3D model is always in-front of the camera. The target body IC pose is varied in five linear directions and seven orientations. Details of the target object position and orientation IC directions are provided in Tables 6.3 and 6.4 respectively. The Z -position is not varied since there is already Z -direction variation in the 3D model misalignment. The error success criteria are in Table 6.5. The success criteria represent less than one percent variation of the input. Primary pose estimation settings are in Table 6.6. The baseline β_0 settings are in Table 6.1. The Sinkhorn [420] normalisation iteration was set to 100 cycles.

Table 6.7 provides the pose estimation results. The original SoftPOSIT results show very low pose estimation success rate, this is because the tight success criteria provided in Table 6.5. Also, for comparability, we do not perform the initial RANSAC search in

Table 6.2: 3D model IC orientation, all angles are in degrees.

Case	X	Y	Z	Roll	Pitch	Yaw
1	0	0	10	0	0	0
2	5	5	15	45	45	45
3	10	10	30	135	135	135
4	-5	-5	5	-45	-45	-45
5	-10	-10	1	-135	-135	-135

Table 6.3: Target object position IC, in length unit [L].

Case	X	Y	Z
1	0	0	5
2	5	0	5
3	0	5	5
4	-5	0	5
5	0	-5	5

Table 6.4: Target object orientation IC, all angles in degrees.

Case	Roll	Pitch	Yaw
1	0	0	0
2	90	0	0
3	0	90	0
4	0	0	90
5	-90	0	0
6	0	-90	0
7	0	0	-90

Table 6.5: Pose estimation error success criteria.

Parameter	Values
Maximum Position Error ([L])	0.05
Maximum Axis Angle Error (deg)	1

Table 6.6: SoftPOSIT enhancement study pose estimation run settings.

Parameter	Values
β final	1000
β increment	1.05
α	1
$s(f)$	1000
$\max r_i - r_{i-1} $	1e-14
$\max \phi_i - \phi_{i-1} $	1e-14

the baseline runs [100]. Both global minimum search, central matching, and combined global minimum search with central matching have significantly improved the pose solution success rate. The highest success is from combined global minimum search with central matching.

Table 6.7: β_0 initialisation by centroid matching results.

Description	Suc- cess	%Suc- cess
Original	12	1
Centroid Matching	30	3
Global Minimum Search	449	51
Global Minimum Search+ Centroid Matching	610	70

David *et al.* [100] propose RANSAC reinitialisation to overcome local minimum traps; this, however, can be costly as David's test show the computation could take from 4 seconds to 6 minutes using 20 to 30 points [100]. If the RANSAC restart is omitted, the baseline SoftPOSIT success rate for a single batch is 131 out of 6,454 cases with a mean time of less than 3 seconds*. Figure 6.10 shows the successful cases as a pose tolerance increased from 1 to 20 degrees and 5 cm to 1 meter. All enhancements except for centroid matching show a constant success rate. Centroid matching shows a steady increase in success as error tolerance increases. Figure 6.10 shows a clear benefit of our proposed enhancement on the pose estimation process. Figure 6.11 shows the average match time as the percentage of

*Simulated in 32 bit-Windows Matlab with 2.4GHz Intel®Core™ 2 Quad Q6600 processor

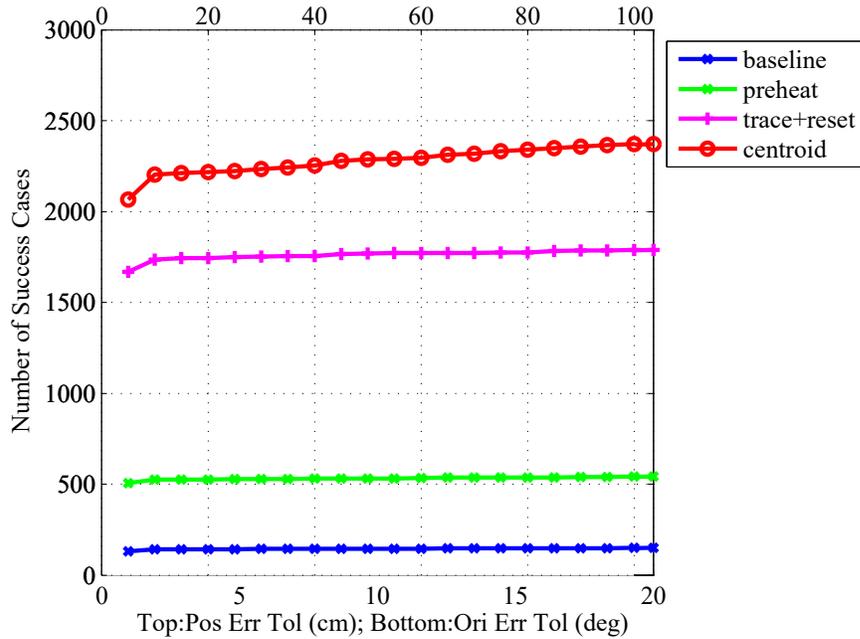


Figure 6.10: Number of successes as a function of error tolerances.

the baseline algorithm; there is almost 50 percent time increase in each enhancement.

Figure 6.12 shows the enhanced SoftPOSIT pose estimation of the RSM given an initial orientation of nearly 123 degrees. The RSM image is captured by the ICI IR camera. The position IC for the 3D model is 0.650, -0.025 , and -0.083 meters in the X , Y , and Z axis respectively, from the \mathcal{F}_{SB} frame to the \mathcal{F}_{CB} frame expressed in the \mathcal{F}_{SB} frame. The orientation IC for the internal model is -62.4 , 105.1 , and -21.3 degrees in *roll*, *pitch*, and *yaw* respectively in the PYR Euler angle rotation sequence, rotating from the \mathcal{F}_{SB} frame to the \mathcal{F}_{CB} frame. The enhanced softPOSIT iteration used 1.031 seconds to complete 43 iterations. The computed RSM pose is 0.686, -0.013 , -0.005 meters and -88.1 , 90.4 , -1.1 degrees. The large IC misalignment increased the run complexity and resulted in longer run time and more iteration steps.

6.9 $ePnP$

Efficient- PnP ($ePnP$) was initially proposed by Moreno-Noguer *et al.* [253] and Lepetit [235] and has been widely used as a fast non-iterative PnP solver. $ePnP$ uses Virtual

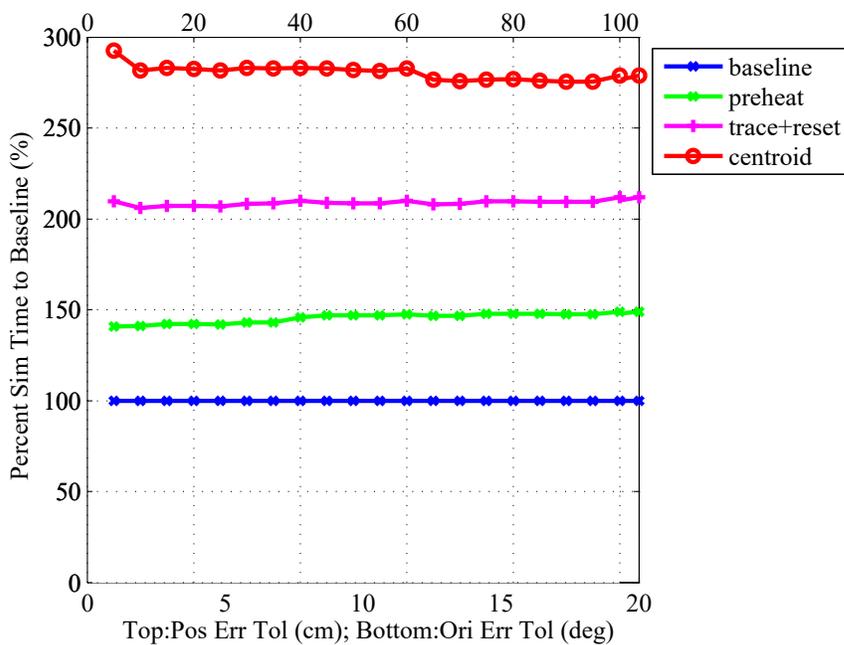


Figure 6.11: Percent iteration time as a function of the error tolerance.

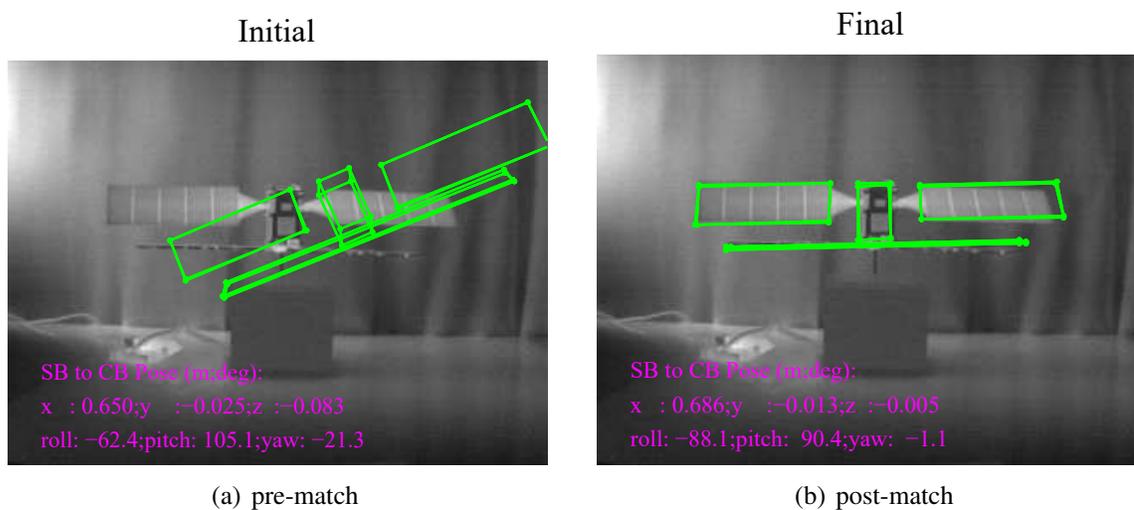


Figure 6.12: SoftPOSIT pose estimation of the RSM captured by an ICI IR camera.

Control Points (VCP) as the basis for all the model Correspondence Points (CP), and provide bounding conditions by using geometric restrictions from inter-VCP distances. An extended version of *ePnP* adds a Gauss-Newton iteration step at the end to increase precision. For a stable solution, at least six points are preferred for the number of correspondence. The *ePnP* procedure begins by taking the centroid of the CP set,

$$\bar{\mathbf{P}}^w = \frac{1}{M} \sum_{k=1}^M \mathbf{P}_k^w. \quad (6.45)$$

The distance of the CP to the centroid is $\mathbf{d}_k^w = \mathbf{P}_k^w - \bar{\mathbf{P}}^w$. The first VCP is the computed centroid, the rest of the VCP are computed based on PCA. Let \mathbf{X} contain all M columns of \mathbf{d}_k^w . The eigenvector and eigenvalues of \mathbf{X} 's covariance matrix is,

$$\begin{aligned} \mathbf{S} &= \mathbf{X}\mathbf{X}^T \\ &= \mathbf{P}_s \mathbf{\Lambda}_s \mathbf{P}_s^T, \end{aligned} \quad (6.46)$$

where \mathbf{e}_i is the eigenvector columns of \mathbf{P}_s , and the diagonal values of $\mathbf{\Lambda}_s$ are the eigenvalues λ_i for $i = 1 \dots 3$. In practice, we use SVD to compute $\sqrt{\lambda_i}$ using Eq. (5.19). The rest of the VCP are defined as,

$$\mathbf{c}_{i+1} = \mathbf{c}_1 + \sqrt{\frac{\sqrt{\lambda_i}}{M}} \mathbf{e}_i. \quad (6.47)$$

Each CP is then the barycentre of the VCP and can be express as,

$$\mathbf{p} = \sum_{i=1}^4 \alpha_i \mathbf{c}_i. \quad (6.48)$$

The α coefficients can be computed as follows,

$$\alpha_1 = 1 - \sum_{i=2}^4 \alpha_i, \quad (6.49)$$

$$\mathbf{A} = \begin{bmatrix} \mathbf{c}_2 - \mathbf{c}_1 & \mathbf{c}_3 - \mathbf{c}_1 & \mathbf{c}_4 - \mathbf{c}_1 \end{bmatrix}, \quad (6.50)$$

$$\begin{bmatrix} \alpha_2 \\ \alpha_3 \\ \alpha_4 \end{bmatrix} = \mathbf{A}^{-1}(\mathbf{x} - \mathbf{c}_1). \quad (6.51)$$

Next, the camera intrinsic properties and the projected image points are used to relate the VCP with the camera and body frames. It can be shown the α coefficients are the same for CP relative to the body and camera frames; this relationship allows geometric restrictions to be formed from both frames and ultimately return the pose of the body frame relative to the camera frame. Substituting the VCP definition from Eq. (6.48) into Eq. (6.6), let $\mathbf{x}^{cT} = \begin{bmatrix} \mathbf{c}_1^{cT} & \mathbf{c}_2^{cT} & \mathbf{c}_3^{cT} & \mathbf{c}_4^{cT} \end{bmatrix}$ where c denotes relative to and express in the camera frame,

$$\mathbf{M}_k = \alpha_k \otimes \begin{bmatrix} fS_x & 0 & \tilde{o}_x - u_k \\ 0 & fS_y & \tilde{o}_y - v_k \end{bmatrix}, \quad (6.52)$$

where \mathbf{M} consists of $2M$ rows from \mathbf{M}_k , and \otimes is the Kronecker product. Next, the following equation can be derived,

$$\mathbf{M}\mathbf{x}^c = \mathbf{0}. \quad (6.53)$$

Computing the SVD of $\mathbf{M}\mathbf{M}^T$,

$$\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T = \mathbf{M}\mathbf{M}^T, \quad (6.54)$$

\mathbf{x}^c is the linear combination of the last four singular vectors of \mathbf{U} as follows,

$$\mathbf{x}^c = \sum_{l=1}^4 \beta_l \mathbf{v}_l. \quad (6.55)$$

Lepetit [235] suggests there are four solution scenarios as $l = 1 \dots 4$ and the order is proportional to the camera focal length. We find the most stable solution comes from the one singular value scenario. To solve for the β coefficient, let $\mathbf{v}_l^T = \begin{bmatrix} \mathbf{v}_{l1}^T & \mathbf{v}_{l2}^T & \mathbf{v}_{l3}^T & \mathbf{v}_{l4}^T \end{bmatrix}$. Six unique geometric distances between the body VCP can be found

$$\rho_r = \|\mathbf{c}_a^w - \mathbf{c}_b^w\|^2, \quad (6.56)$$

where a or b can represent any combination of VCP from 1 to 4. Since the VCP geometric constraint is true for both camera and body frame, the VCP distances are equated as,

$$\|\mathbf{c}_a^w - \mathbf{c}_a^w\| = \|\mathbf{c}_a^c - \mathbf{c}_a^c\|, \quad (6.57)$$

substituting Eq. (6.55) into Eq. (6.56), the following equation can be derived.

$$\mathbf{L}\tilde{\boldsymbol{\beta}} = \boldsymbol{\rho}, \quad (6.58)$$

where $\tilde{\boldsymbol{\beta}}$ is an array containing $\tilde{\beta}_s = \beta_{nm}$, $s = 1 \dots 10$ and $nm = 11, 22, 21, 33, 32, 31, 44, 43, 42, 41$ respectively. Equation (6.58) is the linearisation form of the β polynomial, the double indices indicate

$$\tilde{\beta}_{mn} = \beta_m \beta_n. \quad (6.59)$$

For each of the four cases of l , the general Eq. (6.58) can be simplified. The simplest and most useful case is when $l = 1$ where β is a single value and can be determined as follows,

$$\begin{aligned} \beta &= \frac{\sum_{a,b} \|\mathbf{v}_{1a} - \mathbf{v}_{1b}\| \|\mathbf{c}_a^w - \mathbf{c}_b^w\|}{\sum_{a,b} \|\mathbf{v}_{1a} - \mathbf{v}_{1b}\|^2} \\ &= \sqrt{\frac{\mathbf{L}^T \boldsymbol{\rho}}{\mathbf{L}^T \mathbf{L}}}. \end{aligned} \quad (6.60)$$

For higher l cases, let,

$$\mathbf{d}\mathbf{v}_{abnm} = (\mathbf{v}_{na} - \mathbf{v}_{nb}) \cdot (\mathbf{v}_{ma} - \mathbf{v}_{mb}). \quad (6.61)$$

The matrix \mathbf{L} can be generated by $l_{rs} = \mathbf{d}\mathbf{v}_{abnm}$ entries. The r rows are combinations of the VCP distance constraint, where $r = 1 \dots 6$ are $ab = 12, 13, 14, 23, 24, 34$ respectively. The s columns are the combinations of singular values, where $s = 1 \dots 10$ are the nm combinations previously defined. Once the \mathbf{L} matrix is formed, its SVD can be determined as,

$$\mathbf{U}_l \boldsymbol{\Sigma}_l \mathbf{V}_l^T = \mathbf{L}. \quad (6.62)$$

Equation (6.58) can be solved in the least squares for $l = 2$ and $l = 3$ respectively as,

$$\tilde{\beta}_{l=2} = \mathbf{V}_l \left(\Sigma_l^T \Sigma_l \right)^{-1} \Sigma_l^T \mathbf{U}_l^T \boldsymbol{\rho}, \quad (6.63)$$

$$\tilde{\beta}_{l=3} = \mathbf{V}_l \Sigma_l^{-1} \mathbf{U}_l^T \boldsymbol{\rho}. \quad (6.64)$$

For $l = 4$, the rank needed for the \mathbf{L} matrix inverse exceeds the six constraints from Eq. (6.57). Lepetit [235] proposes a re-linearisation using the properties,

$$\begin{aligned} \tilde{\beta}_{mnpq} &= \beta_m \beta_n \beta_p \beta_q \\ &= \tilde{\beta}_{m'n'p'q'}, \end{aligned} \quad (6.65)$$

where $[m', n', p', q']$ is any combination of $[m, n, p, q]$. The accuracy for the $ePnP$ can be further enhanced by using Gauss-Newton (GN) iteration. The formulation for $ePnP+GN$ is as follows. We define the square error in each of the distance constraint be ϵ_r ,

$$\epsilon_r = \|\mathbf{c}_a^c - \mathbf{c}_b^c\|^2 - \|\mathbf{c}_a^w - \mathbf{c}_b^w\|^2. \quad (6.66)$$

In matrix form, Eq. (6.66) becomes,

$$\begin{aligned} \boldsymbol{\epsilon} &= \boldsymbol{\rho}^c - \boldsymbol{\rho}^w \\ &= \mathbf{L} \tilde{\boldsymbol{\beta}} - \boldsymbol{\rho}^w, \end{aligned} \quad (6.67)$$

the partial rate of change of ϵ with respect to β_l is,

$$\begin{aligned} \frac{\partial \epsilon_r}{\partial \beta_1} &= \left[2L_{r1} \quad L_{r3} \quad L_{r6} \quad L_{r10} \right] \boldsymbol{\beta} \\ \frac{\partial \epsilon_r}{\partial \beta_2} &= \left[L_{r3} \quad 2L_{r2} \quad L_{r5} \quad L_{r9} \right] \boldsymbol{\beta} \\ \frac{\partial \epsilon_r}{\partial \beta_3} &= \left[L_{r6} \quad L_{r5} \quad 2L_{r4} \quad L_{r8} \right] \boldsymbol{\beta} \\ \frac{\partial \epsilon_r}{\partial \beta_4} &= \left[L_{r10} \quad L_{r9} \quad L_{r8} \quad 2L_{r7} \right] \boldsymbol{\beta}, \end{aligned} \quad (6.68)$$

where $\boldsymbol{\beta} = \left[\beta_1 \quad \beta_2 \quad \beta_3 \quad \beta_4 \right]^T$. Collecting $\partial \epsilon_r / \partial \beta_l$, where \mathbf{g}_r is the individual rows of gradient matrix \mathbf{G} , $\mathbf{g}_r = \left[\partial \epsilon_r / \partial \beta_1 \quad \partial \epsilon_r / \partial \beta_2 \quad \partial \epsilon_r / \partial \beta_3 \quad \partial \epsilon_r / \partial \beta_4 \right]$. The GN iteration is then

performed as follows,

$$\beta_{z+1} = \beta_z - \left(\mathbf{G}^T \mathbf{G} \right)^{-1} \mathbf{G}^T \epsilon. \quad (6.69)$$

For real-time operations, the number of iteration steps is cut off at 5 cycles, which in most instances is sufficient for convergence. Once β_l is found, it is possible to extract the body frame pose using the least square method provided by Challis [424] as follows. First find the centroid of the CP from the camera frame

$$\bar{\mathbf{P}}^c = \frac{1}{M} \sum_{k=1}^M \mathbf{P}_k^c, \quad (6.70)$$

the distance of the CP to the centroid is $\mathbf{d}_k^c = \mathbf{P}_k^c - \bar{\mathbf{P}}^c$. Recall the body frame CP is related to the camera frame CP through the extrinsic pose,

$$\mathbf{P}_k^c = \mathbf{t} + \mathbf{R} \mathbf{P}_k^w. \quad (6.71)$$

Substituting the centroid to CP distance for the body and camera CP into Eq. (6.71) generates a *cross-dispersion* matrix \mathbf{C} . Solving the SVD of the cross-dispersion matrix produces the following,

$$\begin{aligned} \mathbf{C} &= \frac{1}{M} \sum_{k=1}^M \mathbf{d}_k^c \mathbf{d}_k^{wT} \\ &= \mathbf{U}_c \boldsymbol{\Sigma}_c \mathbf{V}_c^T. \end{aligned} \quad (6.72)$$

The rotation from the body frame to the camera frame can be computed as,

$$\mathbf{R} = \mathbf{U}_c \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & |\mathbf{U}_c \mathbf{V}_c^T| \end{bmatrix} \mathbf{V}_c^T. \quad (6.73)$$

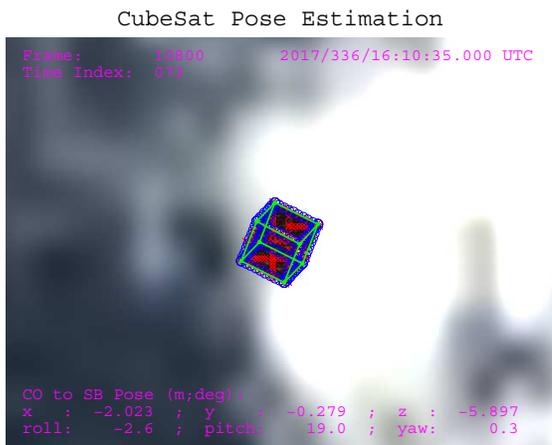
The translation from the camera frame to the body frame expressed in the camera frame is,

$$\mathbf{t} = \bar{\mathbf{P}}^c - \mathbf{R} \bar{\mathbf{P}}^w. \quad (6.74)$$

6.10 Estimation Performance

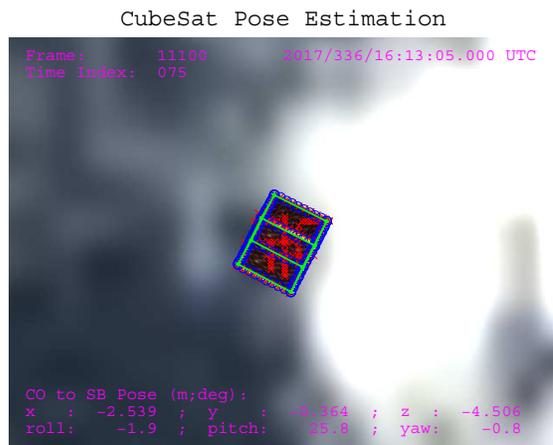
We compared the performances of using edge-line point inflation versus image features and using SoftPOSIT versus $ePnP$. The first example is the CubeSat pose estimation using edge-line point inflation to generate image points and using SoftPOSIT for the PnP solver. The main purpose of this experiment is to show the performance of the edge-line point inflation method. Qualitative representative frames are provided in Fig. 6.13. Quantitative pose comparisons with the ground truth are provided in Fig. 6.14. Frames 10800, 11100, 11960 and 12240 show relatively good pose match, whereas frame 11630 and frame 12090 produced erroneous matches. A closer look into frame 11630 shows the CS far left side edges were not detected and no image points were produced; this caused SoftPOSIT to align itself with the available points which are the incorrect pose. In frame 12090, CubeSat edge was not detected; the SoftPOSIT algorithm aligned the model projection to match the straight lines in the Earth background. Similar to frame 11630, the root cause of pose estimation errors in frame 12090 is due to image processing. In the entire video sequence, the lack of good input image points is the main cause for the pose estimation error. The effectiveness of back-end tools like RANSAC is limited when it comes to compensating for lack of good input image points. The quantitative results in Fig. 6.14 show large deviations from the ground truth which is mainly caused by lack of good input image points.

Our second example is the pose estimation of the Envisat spacecraft undergoing 360 degrees yaw rotation shown in Fig. 6.15. The green outline shows the initial homography mapping, and the red outline shows the final PnP pose estimation. The computation time and pose error performance are provided in Fig. 6.16. The performance figure shows computation time, translation and rotational errors. For small rotation angles, the pose error for both $ePnP$ and SoftPOSIT is comparable; however, the computation timing using SoftPOSIT is by order of magnitude higher than $ePnP$. When the initial misalignment becomes large, the SoftPOSIT method no longer provides a good solution as the maximum limit of the allowed iteration is reached and the program returns without a solution. Given the time required for a single SoftPOSIT iteration, additional RANSAC trials will exceed the non-iterative $ePnP$ timing performance. By comparison, the non-iterative $ePnP$ approach is superior to the iterative approach; the only drawback is it requires more than 6 CP. It is shown in Fig. 6.17, 20 $ePnP$ CP is optimal for image noise resilience. The need for



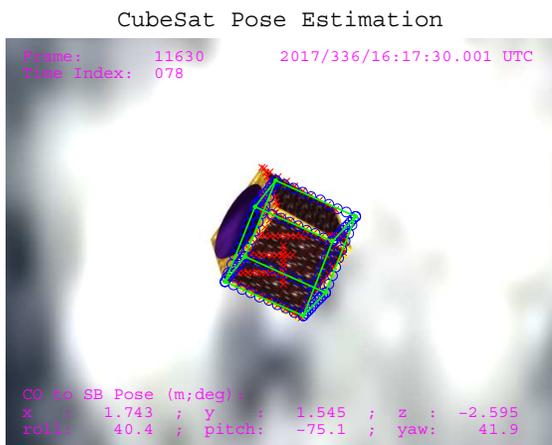
× Detected Corner ○ Predicted Pose

(a) frame 10800



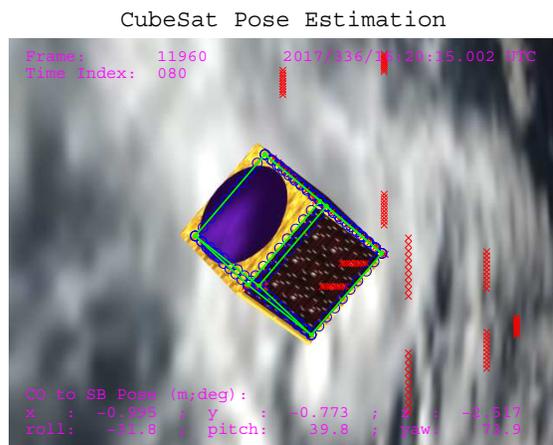
× Detected Corner ○ Predicted Pose

(b) frame 11100



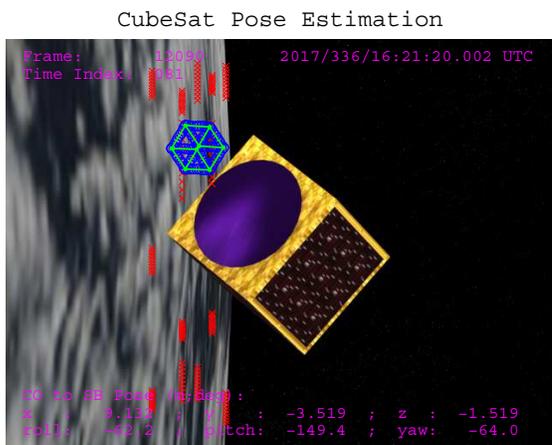
× Detected Corner ○ Predicted Pose

(c) frame 11630



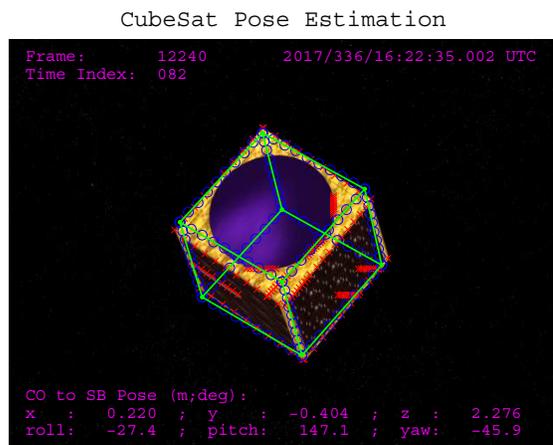
× Detected Corner ○ Predicted Pose

(d) frame 11960



× Detected Corner ○ Predicted Pose

(e) frame 12090



× Detected Corner ○ Predicted Pose

(f) frame 12240

Figure 6.13: Qualitative pose estimation results.

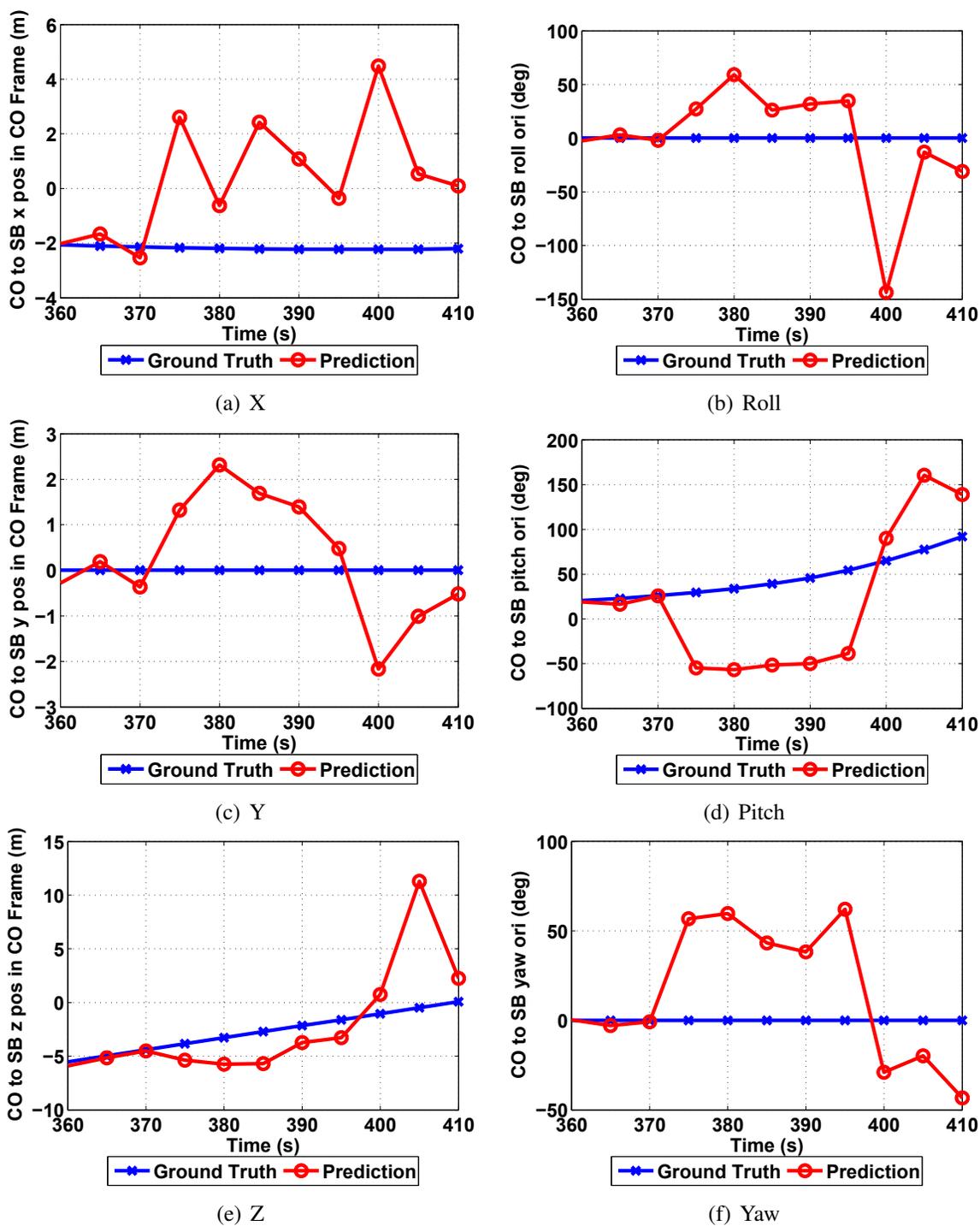


Figure 6.14: Quantitative pose estimation results.

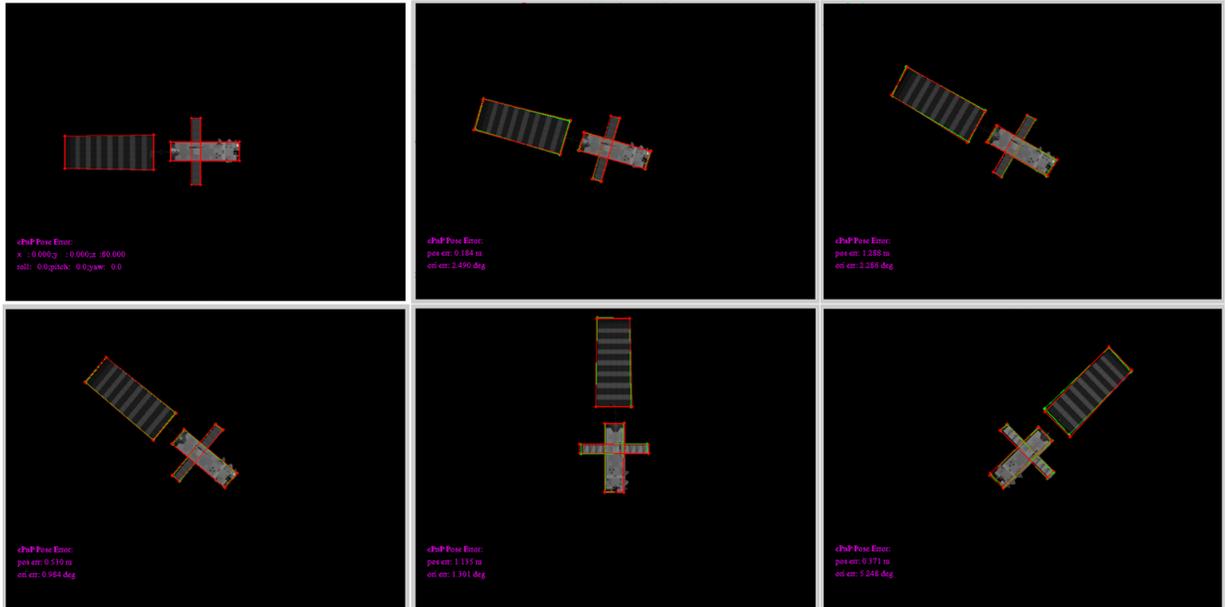


Figure 6.15: Various estimated Envisat pose, the minimum angle is 0 deg, the maximum angle is 135 deg.

20 model points is not a major issue since most spacecraft 3D models will satisfy this requirement. Over the entire video sequence, the most significant source of error is in the generation of the homography matrix. The pose limit sentinel smoothes the output pose; however, there are still many erroneous frames in the estimated sequence. The root cause for the homography matrix generation error can be related to incorrect matches where the RANSAC algorithm does not guarantee the correct template to input correspondence. One could attribute the error to the invariance performance of the image feature. In the next chapter, we will discuss an alternative approach using region-based pose estimation.

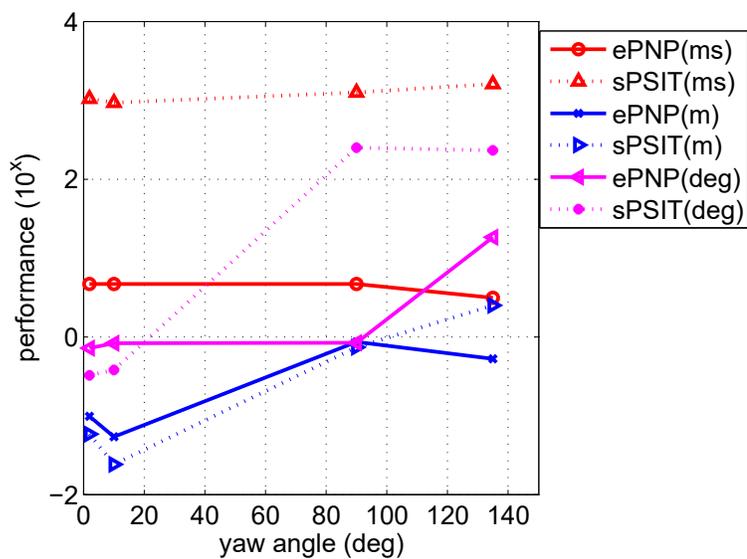


Figure 6.16: *ePnP* vs. SoftPOSIT performances; red lines represent run time, blue lines represent position error, magenta lines represent orientation error.

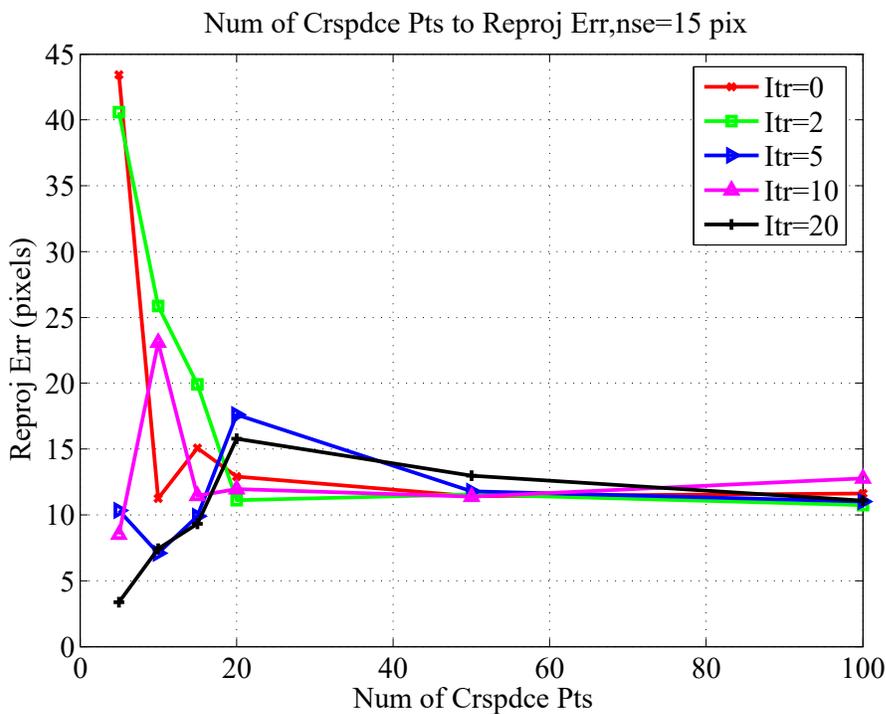


Figure 6.17: Correspondence points vs. image noise for various GN iterations.

Chapter 7

Region-based Method

In this chapter, we provide the region-based formulation for solving the pose determination problem. We also introduce our enhancements in the initialisation and the gradient descent phase of this method. Similar to point correspondence pose estimation, the region-based pose estimation in principal should also perform pose determination without the target spacecraft initial pose. Our testing show, pose determination can be generally achieved with simple spacecraft such as Envisat or Radarsat but is not achievable with complex vehicles like the ISS. In the case of the ISS, *a priori* knowledge of the pose state is needed to limit local minimum trapping in the gradient descent process. Comparatively, the region-based pose estimation is more robust than the point-based approach, and the initial misalignment tolerance can be larger. This chapter is organised in the following sections: Section 7.1 provides the methodology for the level-set segmentation based pose estimation. Section 7.2 provides our enhanced initialisation and gradient descent methods. Section 7.3 provides details of the images used for evaluation and discussions of the results for the region pose estimation approach.

7.1 PWP3D

We extract the best-estimated foreground image from a complex background according to methods given in Shi *et al.* [300]. The best estimate foreground mask can be used as a prior or to enhance a known prior input for the level-set pose estimation process. The saliency generated segmentation can also simplify the input image for a region-based pose estimation that is computed by the combined level-set segmentation and 3D model registration. The level-set refinement minimises an energy function using 3D model projection feedback and foreground-background pixel likelihood estimation.

7.1.1 Notation Definitions and Rotation Transformations

Given an input image I and the image domain $\Omega \subset \mathbb{R}^2$. The image pixel \mathbf{x} with coordinates (x, y) has a corresponding feature \mathbf{y} . This feature could be the pixel intensity or the colour vector (e.g. RGB, CIE Lab). Let us define C as the contour around the object of interest. The foreground region segmented by C is Ω_f , the background is Ω_b . For example, Fig. 7.1(a) shows the contour line of the Envisat image including definitions for its foreground and background regions. The foreground and background regions has its own statistical appearance model, $P(\mathbf{y}|M_i)$ for $i \in \{f, b\}$, where P is the probability density function. Φ is the level-set embedding function. More details of Φ shall be provided in Sec. 7.1.2. Finally, let $H(z)$ and $\delta(z)$ denote the smoothed Heaviside step function and the smoothed Dirac delta function respectively.

A 3D point $\mathbf{X}_c \in \mathbb{R}^3$ with coordinates $(X_c, Y_c, Z_c)^T$ expressed in the camera frame $\vec{\mathcal{F}}_c$ can be a transformation of the object point $\mathbf{X}_b \in \mathbb{R}^3$ expressed in the object body frame $\vec{\mathcal{F}}_b$ with coordinates $(X_b, Y_b, Z_b)^T$. Using a rotation from $\vec{\mathcal{F}}_b$ to $\vec{\mathcal{F}}_c$ denoted by $\mathbf{R} \in SO(3)$ and a translation from $\vec{\mathcal{F}}_c$ to $\vec{\mathcal{F}}_b$ expressed in $\vec{\mathcal{F}}_c$ denoted by $\mathbf{t} = (t_x, t_y, t_z)^T \in \mathbb{R}^3$, where $SO(3)$ is known as the *Special Orthogonal Lie Group*. The rotation matrix, \mathbf{R} , can be parameterised by the quaternion $\mathbf{q} = (q_x, q_y, q_z, q_w)^T$, such that

$$\mathbf{R} = \left(\eta^2 - \boldsymbol{\epsilon}^T \boldsymbol{\epsilon} \right) \mathbf{1} + 2\boldsymbol{\epsilon} \boldsymbol{\epsilon}^T - 2\eta \boldsymbol{\epsilon}^\times, \quad (7.1)$$

where $\mathbf{1} \in \mathbb{R}^{3 \times 3}$ is the identity matrix. Note the sign direction of q_w from Eq. (7.1) for the implementation under this investigation. The individual coordinates of $(\mathbf{t}^T, \mathbf{q}^T)^T$ is represented by λ_i where $i \in \{1, \dots, 7\}$. The transformation from the pitch-yaw-roll Euler angle rotation sequence to quaternion is

$$\begin{bmatrix} q_x \\ q_y \\ q_z \\ q_w \end{bmatrix} = \begin{bmatrix} s_1 c_{23} + c_1 s_{23} \\ s_2 c_{13} + c_2 s_{13} \\ s_3 c_{12} - c_3 s_{12} \\ c_1 c_{23} - s_1 s_{23} \end{bmatrix}, \quad (7.2)$$

where $\mathcal{G}_{ij} = \mathcal{G}\left(\frac{\theta_i}{2}\right)\mathcal{G}\left(\frac{\theta_j}{2}\right)$, $\mathcal{G}_i \in \{c_i = \cos(\alpha_i), s_i = \sin(\alpha_i)\}$, and $i, j \in \{1, 2, 3\}$ for *roll*, *pitch*, and *yaw* angles respectively. A direct transformation from quaternion to the Euler

angles in the pitch-yaw-roll rotation sequence is

$$\begin{bmatrix} \theta_{roll} \\ \theta_{pitch} \\ \theta_{yaw} \end{bmatrix} = \begin{bmatrix} -\text{atan2}(2(q_y q_z - q_x q_w), 1 - 2(q_x^2 + q_z^2)) \\ -\text{atan2}(2(q_x q_z - q_y q_w), 1 - 2(q_y^2 + q_z^2)) \\ \text{asin}(2\sigma) \end{bmatrix}, \quad (7.3)$$

where $\sigma = q_x q_y + q_z q_w$. To handle north and south pole singularity, we check if $|\sigma| > 0.4999$, if satisfied, then the Euler angle rotation is

$$\begin{bmatrix} \theta_{roll} \\ \theta_{pitch} \\ \theta_{yaw} \end{bmatrix} = \begin{bmatrix} 0 \\ \frac{2(q_x q_z + q_y q_w)}{1 - 2(q_x^2 + q_y^2)} \\ \text{sign}(\sigma) \frac{\pi}{2} \end{bmatrix}. \quad (7.4)$$

Finally, the camera is pre-calibrated by the intrinsic matrix

$$\mathbf{K} = \begin{bmatrix} fS_x & fS_\theta & o_x \\ 0 & fS_y & o_y \\ 0 & 0 & 1 \end{bmatrix}, \quad (7.5)$$

where f is the focal length, S_θ is the pixel skew scaling, S_i and o_i are the image scale and centre coordinate for $i \in \{x, y\}$ respectively. Figure 7.1(b) shows the target spacecraft body frame, the camera frame, and the image frame with various position vector and rotation definitions.

7.1.2 Level-set Pose Estimation

The level-set formulation [271] provide a simple mathematical framework for the implicit description of contour evolution. The merging, splitting, appearing, and disappearing of contours can be easily described by a higher dimensional entity Φ than by the explicit formulation of the curve entity C . The contour can be express explicitly as the 0^{th} level in the level-set function Φ . For example, a contour in a two-dimensional image is defined by the zero level in a Lipschitz continuous function Φ in a three-dimensional surface. Formally, $C = \{(x, y) \in \Omega | \Phi(x, y) = 0\}$. An example of the level-set function and the outer contour is shown in Fig. 7.1(c), where the function's 0^{th} -level corresponds to the contour

line from Fig. 7.1(a) and is highlighted in magenta. In Fig. 7.1(c), the level-set function preserves positive and negative parts so it can be clearly illustrated. The level-set function Φ is evolved rather than directly evolving C . The subset of the level-set function is a *signed distance function* $d(\mathbf{x})$ defined as $d(\mathbf{x}) = \min_{\mathbf{x}_i \in C} |\mathbf{x} - \mathbf{x}_i|$, hence, $|\nabla \Phi(x, y)| = 1$. An illustration of the level-set function evolution based on target object motion is provided in Fig. 7.2. The top 2 rows in Fig. 7.2 provide synthetic images of ISS motion with internal model mesh projection overlay. The bottom 2 rows in Fig. 7.2 provide the corresponding level-set functions. The zero-level and zero crossing are indicated by gradient lines and magenta lines respectively. The zero-boundary is also projected above the zero-level as blue lines for clear illustration.

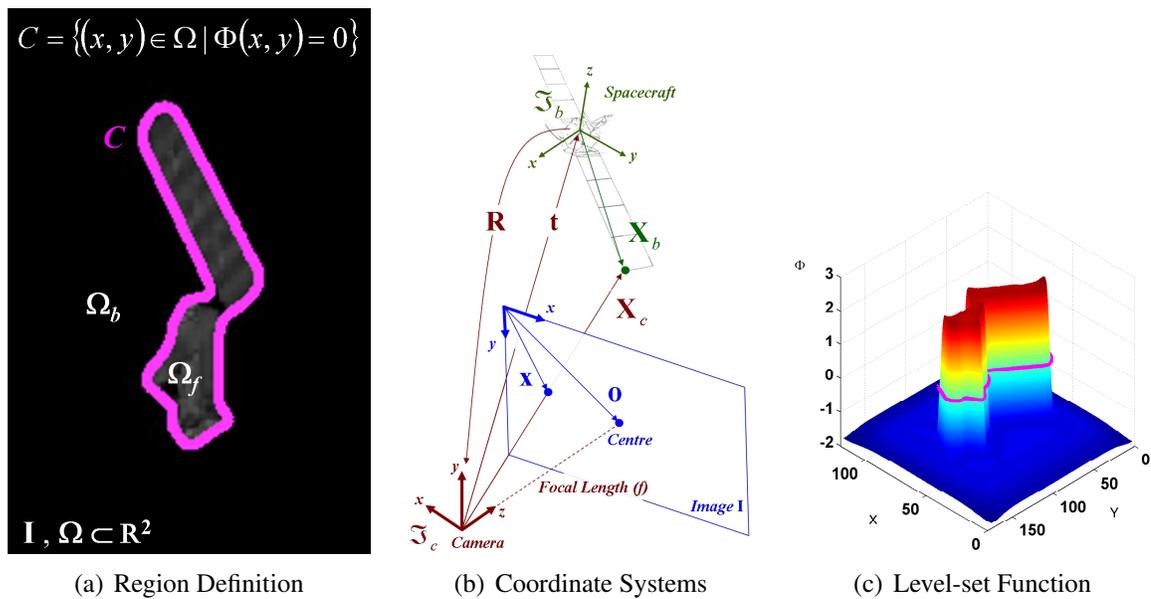


Figure 7.1: Spacecraft image contour, camera coordinate system definition, and the level-set function.

Segmentation Energy

The level-set formulation of the piecewise constant Mumford-Shah functional [268, 270, 425] that produces the two-phase segmentation of an image $I : \Omega \rightarrow \mathbb{R}$ by minimising an

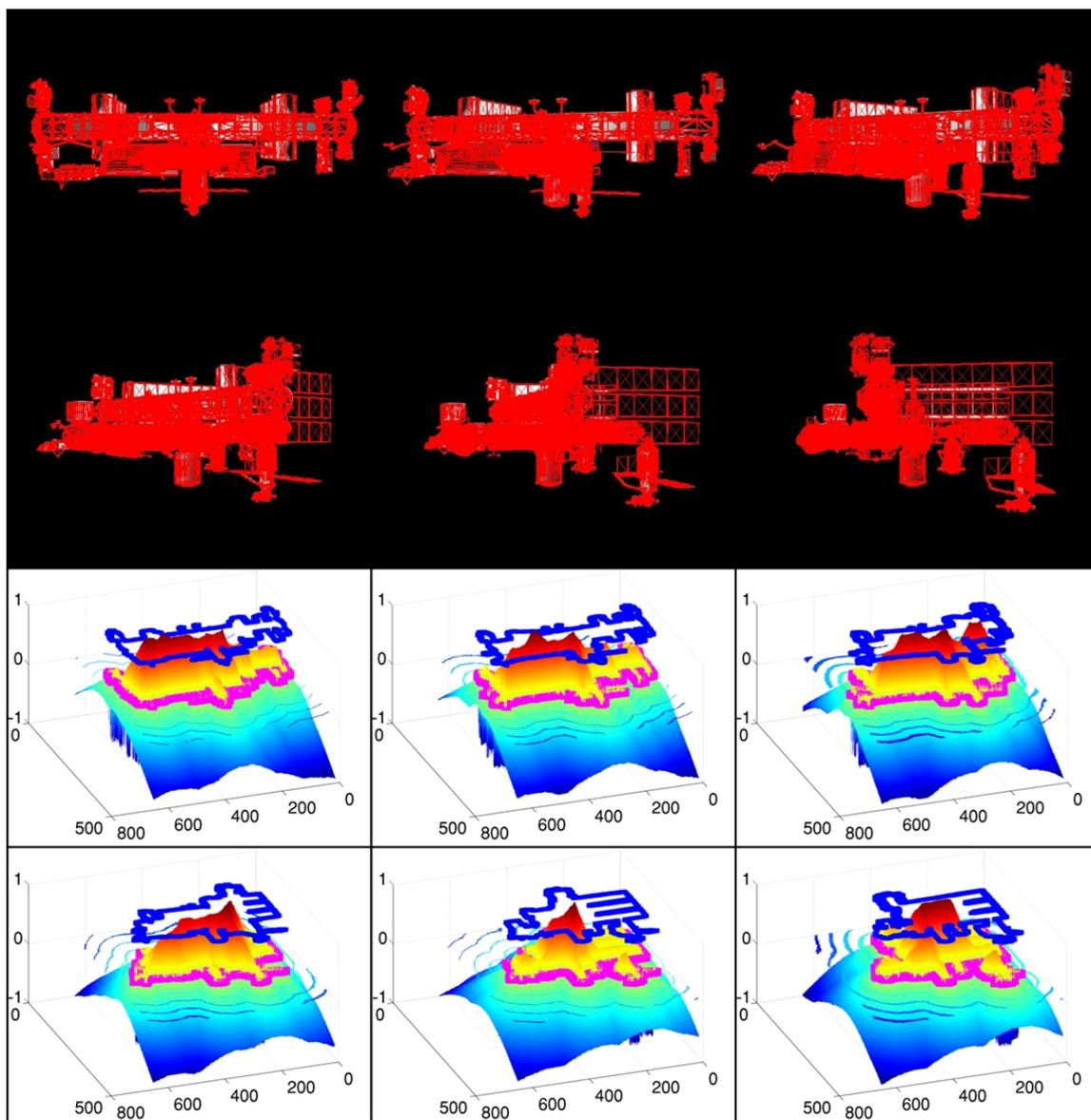


Figure 7.2: ISS pose estimation level-set functions.

energy function [426] given by,

$$\begin{aligned} E &= \int_{\Omega_f} r_f(\mathbf{I}(\mathbf{x}), C) d\Omega + \int_{\Omega_b} r_b(\mathbf{I}(\mathbf{x}), C) d\Omega \\ &= \int_{\Omega} (H(\Phi) r_f(\mathbf{x}) + (1 - H(\Phi)) r_b(\mathbf{x})) d\Omega, \end{aligned} \quad (7.6)$$

where r_i for $i \in \{f, b\}$ is the image property that will be discuss in the following section.

Pixel Likelihood

Taking r_i as the likelihood of the pixel property, where $r_i(\mathbf{x}) = P(\mathbf{y}|M_i)$ and $i \in \{f, b\}$. Bibby and Reid [279] proposed an effective energy formulation as the posterior of each pixel's respective membership. Assuming pixel-wise independence, and replacing the integration with a summation of the log posterior probability of the contour. The energy becomes,

$$\begin{aligned} E(\Phi) &= -\log(P(\Phi|\mathbf{I})) \\ &= -\log\left(\prod_{\mathbf{x} \in \Omega} (H(\Phi) P_f + (1 - H(\Phi)) P_b)\right) \\ &= -\sum_{\mathbf{x} \in \Omega} \log(H(\Phi) P_f + (1 - H(\Phi)) P_b) \end{aligned} \quad (7.7)$$

and the foreground and background probabilities P_f and P_b are

$$P_f = \frac{P(\mathbf{y}|M_f)}{P(\mathbf{y}|M_f) P(M_f) + P(\mathbf{y}|M_b) P(M_b)} \quad (7.8)$$

$$P_b = \frac{P(\mathbf{y}|M_b)}{P(\mathbf{y}|M_f) P(M_f) + P(\mathbf{y}|M_b) P(M_b)} \quad (7.9)$$

where $P(M_i)$ such that $i \in \{f, b\}$, is the prior and can be computed by taking the areas of the respective regions,

$$P(M_f) = \sum_{\mathbf{x} \in \Omega} H(\Phi(\mathbf{x})) \quad (7.10)$$

$$P(M_b) = \sum_{\mathbf{x} \in \Omega} (1 - H(\Phi(\mathbf{x}))) \quad (7.11)$$

3D Model Projection and Pose Estimation

The target object pose can be estimated using the energy functional as described in Eq. (7.7) by taking the partial derivative with respect to the individual pose parameters γ_i ; this allows the evolution of the target boundary with-respect-to its pose rather than time. Let us define $\partial(a)/\partial\gamma_i = a_{\gamma_i}$, $\nabla_t(a) = (a_{t_x}, a_{t_y}, a_{t_z})^T$, and $\nabla_q(a) = (a_{q_x}, a_{q_y}, a_{q_z}, a_{q_w})^T$. The energy partial-derivative is

$$E_{\gamma_i} = - \sum_{\mathbf{x} \in \Omega} \frac{P_f - P_b}{(P_f - P_b) H(\Phi) + P_b} \delta(\Phi) (\nabla \Phi)^T \mathbf{x}_{\gamma_i}, \quad (7.12)$$

where ∇ is the image gradient over \mathbf{x} . The camera projection model can be used to relate the 3D model to the 2D image as follows,

$$\begin{bmatrix} \mathbf{x} \\ 1 \end{bmatrix} = Z_c^{-1} \mathbf{K} \begin{bmatrix} \mathbf{1} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \mathbf{X}_b = \begin{bmatrix} fS_x & fS_\theta & o_x \\ 0 & fS_y & o_y \\ 0 & 0 & 1 \end{bmatrix} \frac{\mathbf{X}_c}{Z_c}, \quad (7.13)$$

where $\mathbf{0} \in \mathbb{R}^3$, \mathbf{K} is the intrinsic camera matrix, \mathbf{X}_c/Z_c is the depth normalised object point observed and expressed from the camera frame, f is the focal length of the camera, S_θ is the pixel skew scaling, S_i and o_i where $i \in \{x, y\}$ is the pixel scaling and image origin to centre distance respectively. Equation (7.13) can be used to derive an expression for \mathbf{x}_{γ_i} such that,

$$\mathbf{x}_{\gamma_i} = \frac{f}{Z_c^2} \begin{bmatrix} \mathbf{X}_c^T (S_x \mathbf{T}_x + S_\theta \mathbf{T}_y) \\ \mathbf{X}_c^T (S_y \mathbf{T}_y) \end{bmatrix} (\mathbf{X}_c)_{\gamma_i}, \quad (7.14)$$

where

$$\mathbf{T}_x = \begin{bmatrix} 0 & 0 & -1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}, \quad \mathbf{T}_y = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}, \quad (7.15)$$

The partial derivative of \mathbf{X}_c with-respect-to the pose parameters γ_i is derived from the extrinsic translation and rotation of the body coordinates to the camera coordinates through $\mathbf{X}_c = \mathbf{R}\mathbf{X}_b + \mathbf{t}$, the partial derivative results are as follows,

$$\nabla_t \mathbf{X}_c^T = \mathbf{1}, \quad \nabla_q \mathbf{X}_c^T = 2 \begin{bmatrix} \mathbf{A}\mathbf{X}_b & \mathbf{B}\mathbf{X}_b & \mathbf{C}\mathbf{X}_b \end{bmatrix}, \quad (7.16)$$

where

$$\mathbf{A} = \begin{bmatrix} 0 & q_y & q_z \\ -2q_y & q_x & -q_w \\ -2q_z & q_w & q_x \\ 0 & q_z & -q_y \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} q_y & -2q_x & q_w \\ q_x & 0 & q_z \\ -q_w & -2q_x & q_y \\ -q_z & 0 & q_x \end{bmatrix}, \quad \mathbf{C} = \begin{bmatrix} q_z & -q_w & -2q_x \\ q_w & q_z & -2q_y \\ q_x & q_y & 0 \\ q_y & -q_x & 0 \end{bmatrix}, \quad (7.17)$$

7.2 Region-based Method Enhancements

The following sections provide region-based method gradient descent convergence speed increase by centre initialisation and modifications to the gradient descent equation.

7.2.1 Centre Initialisation

Section 7.1.2 provide the connection between the model projection and level-set function using pixel probability. The gradient landscape surrounding the final pose minimum depends on the foreground and background pixel intensity variation. If the initial condition pose is specified in a region with black space background far from the target object, the gradient descent process can be highly sluggish. We develop a novel and simple initialisation scheme to avoid black space projection and reduce the number of steps that are required to reach the final pose potential minimum. Let us define the unaltered initial condition pose translation as \tilde{t} , and our altered approach as t . Figure 7.4 shows the original initial condition and our altered approach by using centralisation. The top figure in Fig. 7.4 shows the raw initial condition pose, captured image and actual target in grey. The bottom figure in Fig. 7.4 shows centralised initial condition pose using saliency mask geometric centroid (blue dot) to set the internal model body frame projection. We use the computed saliency mask to generate a geometric centre of the RoI. The pose translation image coordinate is set to the geometric centre, (\bar{x}, \bar{y}) , which is computed using image areal moments. While there is no guarantee the body frame, which is normally the centre of mass, is the saliency mask geometric centre. It is likely this approximation will overlap the RoI than an arbitrary chosen initial pose with a large misalignment. This centre shift allows the gradient descent method to initiate in a region where the pose potential is more pronounced than if

projecting the initial pose into a black space region with even gradients. The initial pose is computed as,

$$\mathbf{t} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \frac{1}{fS_x} (Z_c (\bar{x} - o_x) - fS_\theta y) \\ \frac{Z_c}{fS_y} (\bar{y} - o_y) \\ Z_c \end{bmatrix}. \quad (7.18)$$

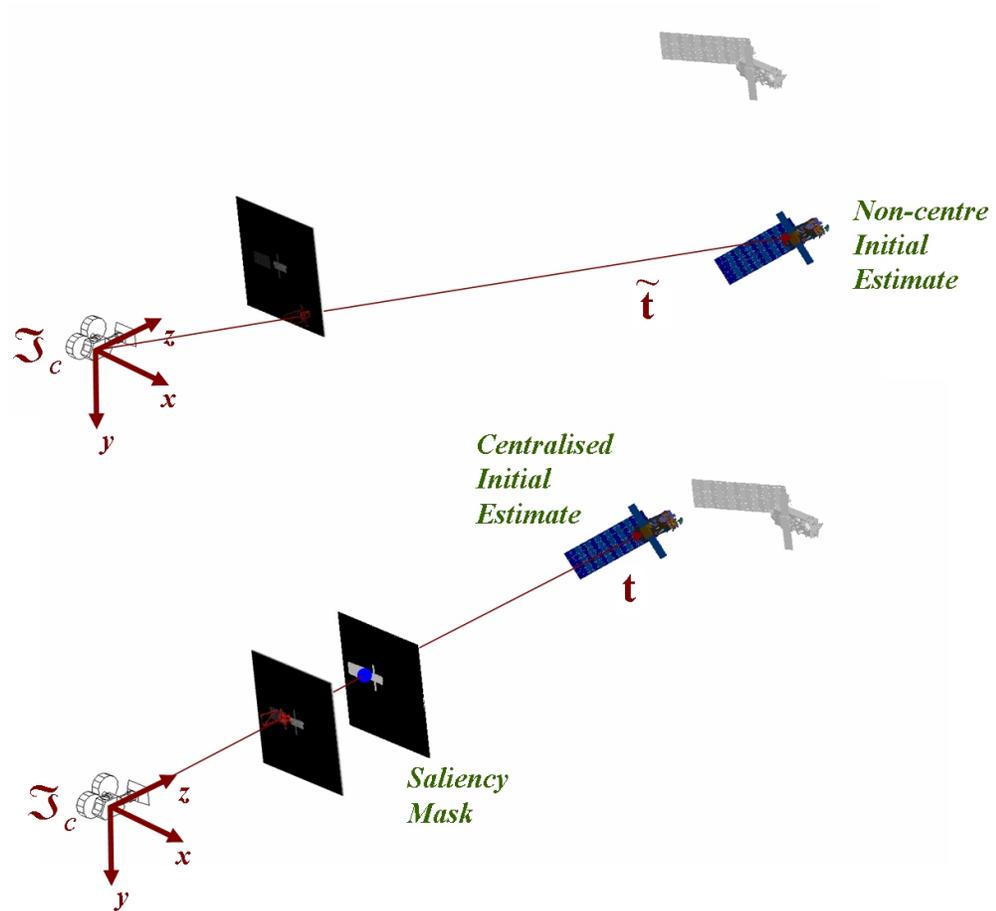


Figure 7.3: Centre initialisation of the target object.

7.2.2 Gradient Descent

The PWP3D [101] gradient descent method increment the pose parameter by manually adjusted stepsize. Let us define the pose vector as $\mathbf{x} = \begin{bmatrix} \mathbf{t}^T & \mathbf{q}^T \end{bmatrix}^T$. The baseline PWP3D

gradient descent is,

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{h} \circ \mathbf{f}(\mathbf{x}_k), \quad (7.19)$$

where \mathbf{h} is the step size for the individual pose axis, \circ is the element-wise multiplication Hadamard product operator, $\mathbf{f}(\mathbf{x})$ is the gradient from Eq. (7.14) and q_{k+1} is L2-normalised after computing Eq. (7.19). The baseline gradient descent procedure is unstable and fails to produce the correct pose in several test cases. We tested various gradient descent procedures including Nelder-Mead multi-dimensional simplex method [427], the combined Polack-Ribiere and Fletcher-Reeves method [101, 428]; however, neither method produced satisfactory timing and accuracy. Finally, we developed an enhanced gradient descent procedure with superior estimation results. The gradient magnitude variation directs the Nelder-Mead simplex [427] state change. The Polack-Ribiere and Fletcher-Reeves [101] step direction include the current and previous step gradients. Our tests show the current gradient direction outperforms both methods and is computationally more efficient and straightforward to implement. Unlike the baseline PWP3D, we use an alternative magnitude based on the inverse of the translational distance magnitude. We use the inverse translational distance to modify the stepsize because closer distance results in a larger image projection and therefore require smaller gradient descent movement and vice-versa. Our improved gradient descent formulation is as follows,

$$\mathbf{t}_{k+1} = \mathbf{t}_k + \frac{h_t}{\|\mathbf{t}\|} \hat{\mathbf{f}}_t(\mathbf{x}_k), \quad (7.20)$$

where $\hat{\mathbf{f}}_t(\mathbf{x}_k)$ is the unit direction of the translational gradient, $\mathbf{f}_t/\|\mathbf{f}_t\|$, and h_t is the translational stepsize. The rotational gradient descent formulation is

$$\tilde{\mathbf{q}}_{k+1} = \mathbf{q}_k + \frac{h_q}{\|\mathbf{t}\|} \hat{\mathbf{f}}_q(\mathbf{x}_k), \quad (7.21)$$

where $\tilde{\mathbf{q}}$ is a non-normalised quaternion, and $\hat{\mathbf{f}}_q(\mathbf{x}_k)$ is the unit vector of the quaternion gradient. The final state vector is $\mathbf{x}_{k+1} = \left[\mathbf{t}_{k+1}^T \quad \tilde{\mathbf{q}}_{k+1}^T / \|\tilde{\mathbf{q}}_{k+1}\| \right]^T$. Figure 7.4 provides the difference of using the basic gradient descent formulation and our enhanced method. Figure 7.4 row 1-2 provides Envisat pose overlay for frame 0, 150, and 300. Figure 7.4 row 3-4 provides RSM pose overlay for frame 200, 250, and 300. Figure 7.4 row 5-6

provides ISS pose overlay for frame 250, 350, and 450. Figure 7.4 row 1, 3, and 5 uses the PWP3D gradient descent implementation [101], row 2, 4, and 6 uses the enhanced gradient descent method. The original version destabilised when the projection silhouette transitioned into the minimum region. The enhanced method remained stable through-out the entire estimation process.

7.3 Estimation Results

We use a combination of synthetic CAD images and space flight infrared images for experiment and evaluation. The CAD images include 3D models of the Envisat spacecraft, the RSM, and the ISS model in basic element shapes and complex meshes. The RSM is a miniature scale laboratory imitation of the Radarsat spacecraft. The 3D models are also the pose estimation software internal models. We use *3D Studio Max*[®] to generate synthetic videos from the 3D models which include some lighting and shadowing effects. The STS-135 ISS docking and undocking mission phases were recorded by a Neptec *TriDAR* thermal camera installed on the SSO. The *TriDAR* thermal image resolution is 813×604 ; for faster processing, the resolution was reduced to 320×240 with estimated camera calibration properties of $fS_x = 752.517$, $fS_y = 752.517$, and $fS_\theta = 0$. Media extensions and data resources of the investigated cases are available on our project website^a.

7.3.1 Stepsize Sequence

A sequence of gradient descent for the CubeSat pose estimation is shown in Fig. 7.5. Estimation typically starts with a large initial condition offset, and subsequent tracking is performed using the segmentation mask from the previous frame to define the foreground and background pixels. A process for step-size tuning was developed for efficient convergence from large initial misalignments. This iteration process is described in detail per Fig. 7.5.

^a<http://ai-automata.ca/research/hisafe.html>

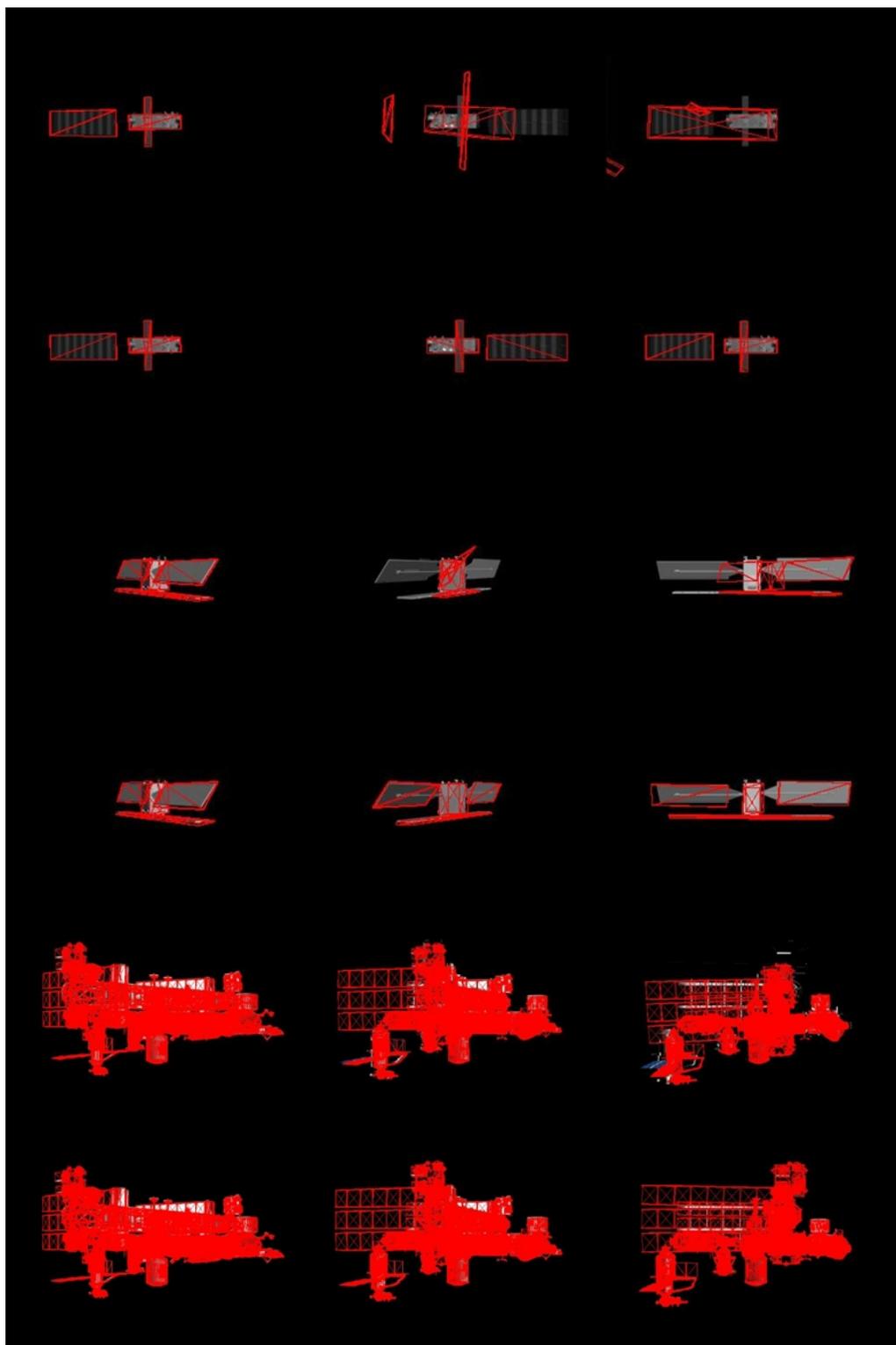


Figure 7.4: The enhanced gradient descent results for Envisat (rows 1-2), RSM (rows 3-4), and ISS (rows 5-6) synthetic image pose estimation.



Figure 7.5: Gradient descent results for a single image. For large initial offset between the model projection and the image, the gradient descent is performed by a coarse depth and lateral movement, followed by rotation (as shown by the first row), then by the same combination in fine step adjustments (as shown by the second row). Finally, even smaller step adjustments in all directions are applied simultaneously (as shown by the third row).

7.3.2 A Priori Enhancement

For each pose generation, the pixel foreground, and background probability posterior are computed and converted to a segmentation mask for the subsequent frame. For normal operations, statistical training is unnecessary for every frame; however, the likelihood needs to be refreshed periodically such that it is current to the observed scenery. The foreground target and the background are usually separable for colour images and images with dark space as the background. Grayscale images with relatively similar foreground and background contrast are much more difficult to classify using pixel distribution of the global image. For images with Earth backgrounds, contour error will creep in over time resulting in pose estimation failure. A remedy for reducing the error creep is to apply Otsu thresholding to the posterior map removing low probability regions and noise; this will result in a clean mask for the subsequent frame level-set estimation as shown in Fig. 7.6.

Degradation of the probability mask can occur if the sequential mask generated by the posterior is replaced with the first-frame *a priori* as input. This is because the training data histogram is corrupted by mixing foreground and background pixels from the wrong pose

projection segmentation mask. This error exposes the primary weakness in using the level-set pose estimation method since it is strongly dependent on having current and separable pixel histograms between the foreground and the background. The Otsu thresholding of the posterior mask is further enhanced with an image fill process to reduce the estimation degradation. The improvement process is provided as follows: 1. Perform Otsu thresholding on the classified foreground posterior map in generating the segmentation mask. 2. Flood fill the segmentation mask with its background. 3. Invert the flood fill segmentation mask. 4. Combine the threshold segmentation mask with the inversed flood filled mask using the bitwise OR operation. The enhancement resulted in a cleaner segmentation mask as shown in Fig. 7.7.

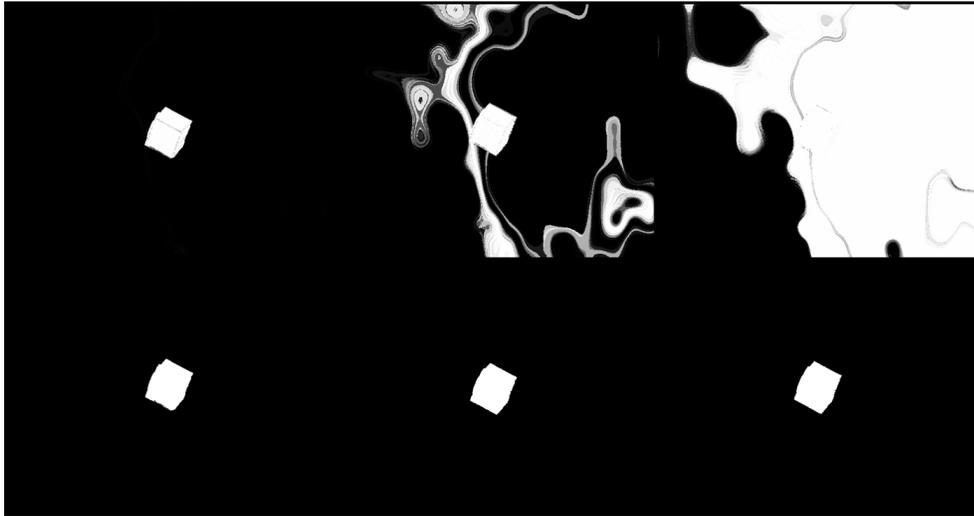


Figure 7.6: Image sequence pose estimation uses the foreground pixel probability posterior generated from the previous frame. Leakage of background contour is amplified after three frames as shown by the first row. The segmentation mask is improved by using the proposed thresholding and fill technique as shown by the second row.

7.3.3 Pose Estimation Error

Multi-sequence pose estimation for the CubeSat is shown in Fig. 7.8. The silhouette projection of the 3D model with enhancements was made to the posterior results to stabilises the level-set contour evolution. While the contour boundary tightly restricts the lateral and depth estimation, rotations of symmetrical bodies such as the CubeSat can have an error drift over the image sequence. This error is accumulated over many frames and can be

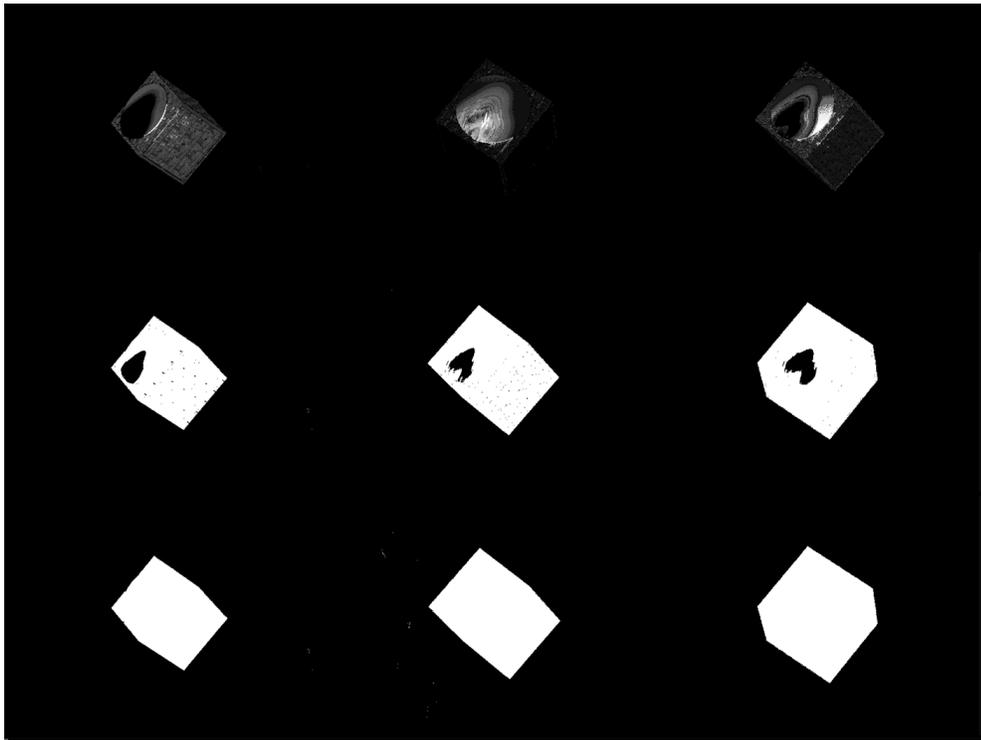


Figure 7.7: The posterior of later frames is degraded over time if only using the *a priori* mask in generating the training histograms (shown by the first row). Using immediate previous frames and Otsu thresholding results in a stronger foreground posterior (shown by the second row). The additional filling of the segmentation mask improves the posterior results (shown by the third row).

difficult to jump out from the local minimum trap. Future developments shall focus on producing better orientation estimations by using PnP techniques internal to the foreground image.

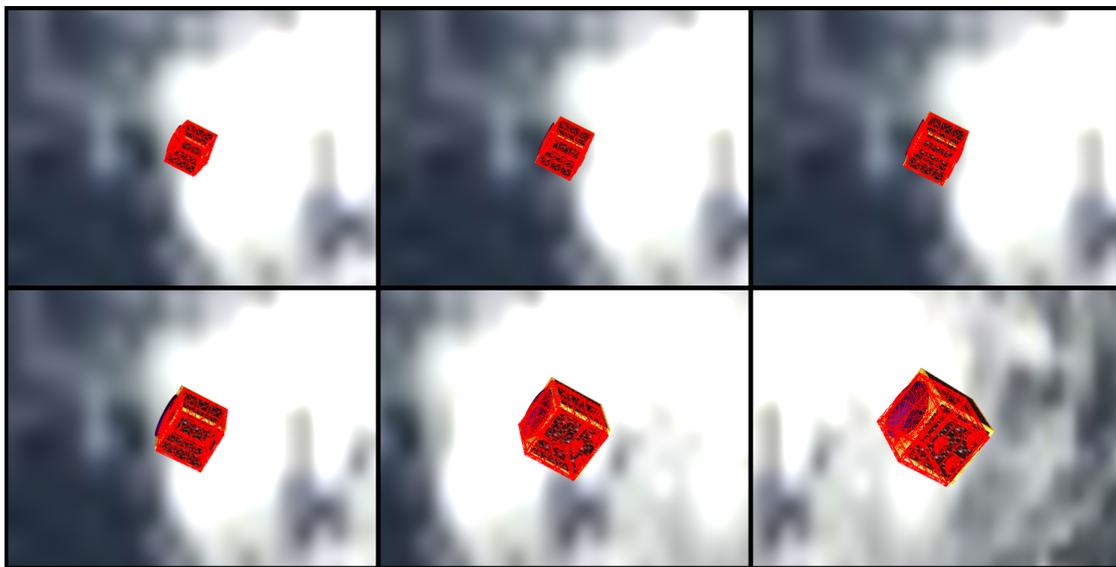


Figure 7.8: Pose estimation results for multiple image frames from an approach maneuver. 3D projection matches the segmented target image with some error in the out-of-plane rotation that is difficult to infer by silhouette registration.

The results of the Envisat rotation sequence and pose estimation are provided in Fig. 7.9. Targets with non-symmetrical geometry has greater resilience to pose iteration and can allow for larger step-size. Results of the RSM rotation sequence and pose estimation are provided in Fig. 7.10. The Root Sum Squared (RSS) error of the position and orientation are provided in Fig. 7.11 for the envisat and radarsat trials from Fig. 7.9 and Fig. 7.10 respectively. Results show the highest error when segmented regions with minimum areas. The RSM exhibit higher relative error due to a combination of large out-of-plane view-point changes, smaller and closer model targets, and greater corruption to the region silhouette due to shadowing. While the core pose parameter for computation uses quaternions, floating point numerical precision near rotational singularities for orientation output displays requires careful attention.

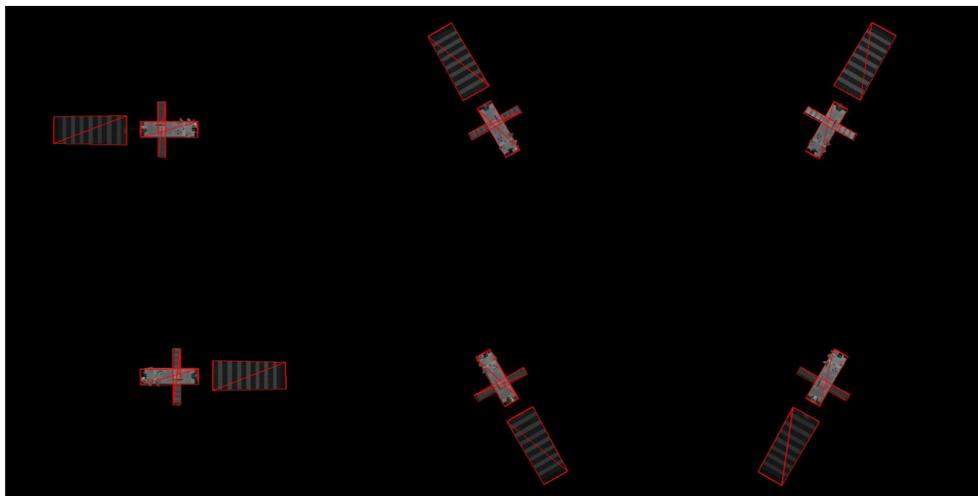


Figure 7.9: Pose estimation results for multiple image frames from an Envisat rotation (<https://youtu.be/8Km--FOmC8E>).

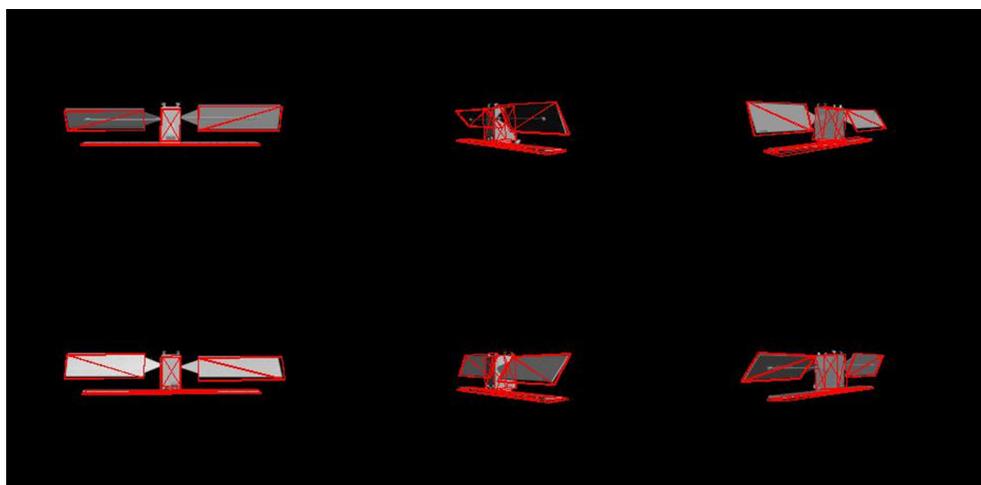


Figure 7.10: Pose estimation results for multiple image frames from a RSM rotation (<https://youtu.be/IEMpdNHJwic>).

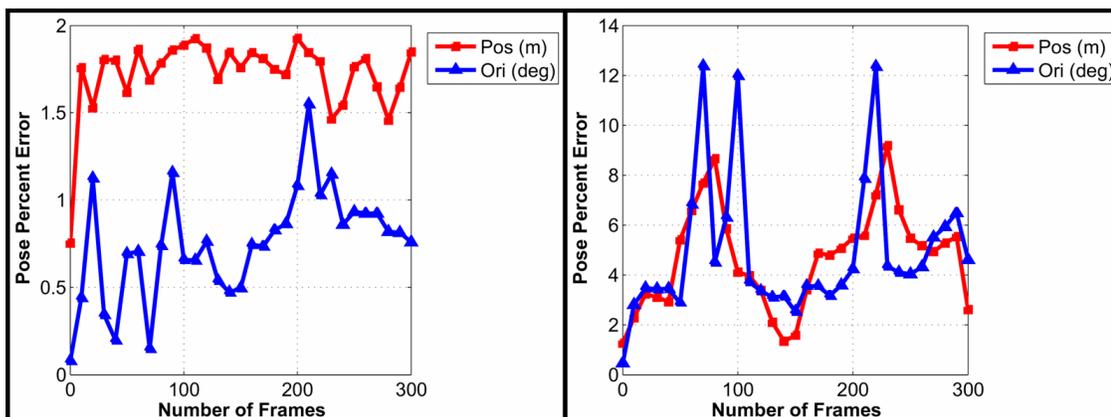


Figure 7.11: Pose estimation percentage error. Left: Envisat trial. Right: RSM trial.

7.3.4 Envisat 6-DOF Pose Estimation

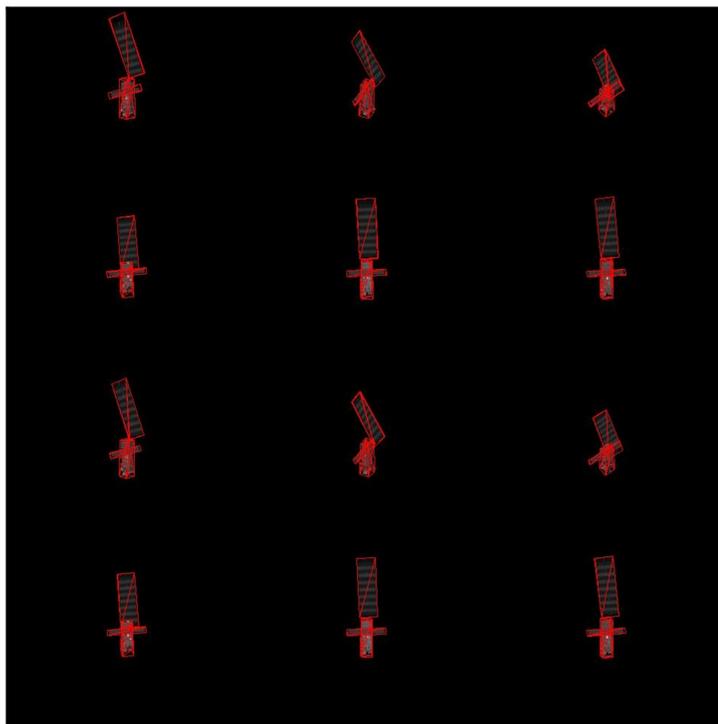
The simulated Envisat motion includes rotation in the roll and yaw axis over two cycles. The Envisat is 80 meters away from the SS camera with a tumbling rate of 10 degrees per second. The camera resolution is 320×240 with calibrated intrinsic properties of $fS_x = 439.967$, $fS_y = 432.427$, and $fS_\theta = -0.0699286$. Figures 7.12 and 7.13 provide the pose estimation of a simulated Envisat tumbling motion. Figure 7.12(a) provides the grayscale image captured by a virtual camera in the simulation environment. The internal simplified 3D model is projected on to the image as visual overlays in red over two tumbling cycles. Figure 7.12(b) provides the equivalent 3D plot of the Envisat motion computed using only the captured images compared to the Ground Truth (GT) in the Envisat spacecraft body frame. The GT is in blue and estimated pose estimation is plotted in red. The camera pose is defined as the position from the camera frame to the Envisat body frame expressed in the camera frame, and orientation of the Envisat body frame rotated from the camera frame using the pitch-yaw-roll Euler angle rotation sequence.

The position and orientation of the Envisat relative to the camera expressed in the camera frame is shown in Figs. 7.13(a) and (b) respectively. The offset error in the X and Y direction is less than 0.1 and 0.5 meters respectively. Out of 80 meters, total distance between the camera and Envisat, the RSS lateral error is 0.64 percent. There is a Z -axis drift of less than 4 meters or 5 percent. It is not surprising the bore sight error is larger than the lateral error while using a single camera to predict the target spacecraft pose. The bore sight prediction is related to the precision of the target areal region and the resolution of the

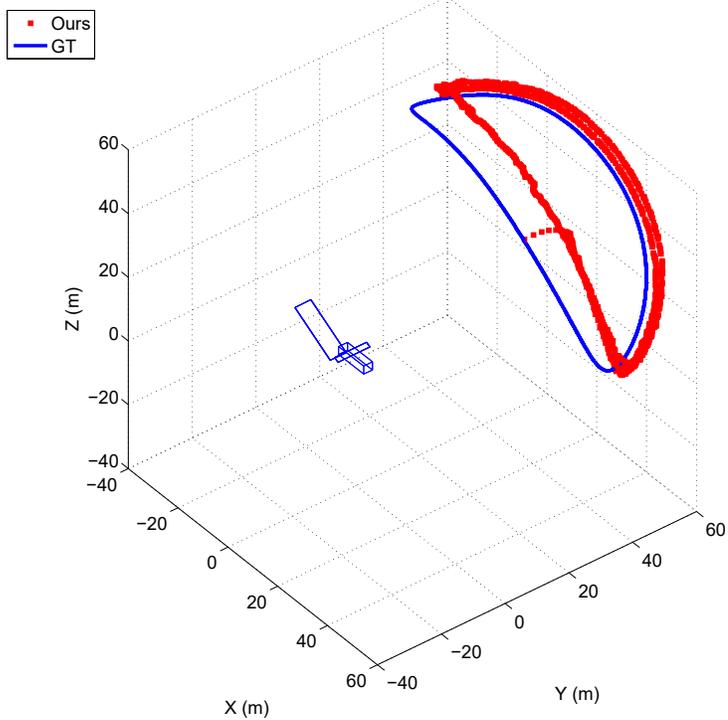
image. A large distance between the target vehicle and the camera will result in a larger bore sight measurement error. An initial translational pose offset shown in Fig. 7.12(b) is caused by an orientation offset when expressed in the camera frame. Figure 7.13(b) shows the orientation offset during peak oscillations. Due to the RSS distance between the two vehicles being 80 meters, any small rotation error will be amplified into larger translation errors when expressed in the Envisat body frame. The orientation offset is largely due to the shadowing of the Envisat lower solar panel and its attachment which reduces the shape of the entire spacecraft region. This error highlights the sensitivity of the region-based pose estimation method to the accuracy in the observable object foreground.

7.3.5 Reduced Model Pose Estimation

Figure 7.14 provides pose estimation results for the synthetic ISS rotation video sequence. The top two rows in Fig. 7.14 provide simplified ISS CAD model rotational motion overlay with simple level-set internal CAD model pose estimation. The middle two rows in Fig. 7.14 include complex ISS CAD model rotational motion overlay with simple level-set internal CAD model pose estimation. The bottom two rows in Fig. 7.14 provide complex ISS CAD model rotational motion overlay with complex level-set internal CAD model pose estimation. The prior for each image in the sequence is the posterior mask generated from the previous image. A combination of a simple internal model to a simple input image, simple internal model to a complex input image, and a complex internal model to a complex input image over 90 degrees yaw rotation is provided in rows 1, 2, rows 3, 4, and rows 5, 6 respectively. In the simple-to-simple case, estimation begins to degrade in the final two images. The input image contains shadows near the radiation panels that resulted in a falsely lit region in the input image; this is the primary cause for the orientation error in the estimated pose. In the simple-to-complex case, the internal model is a coarse version of the input image yet pose estimation remained stable until the 5th image. The same shadowing effects observed in the simple-to-simple case amplified the initial error caused by the model difference. In the complex-to-complex case, the pose estimation matched throughout the entire sequence even with shadowing on the vertical radiation panel. The complex-to-complex case is more robust in pose error than the simple-to-simple case due to higher resolution edge shapes creating a more distinctive histogram profile. Additionally,

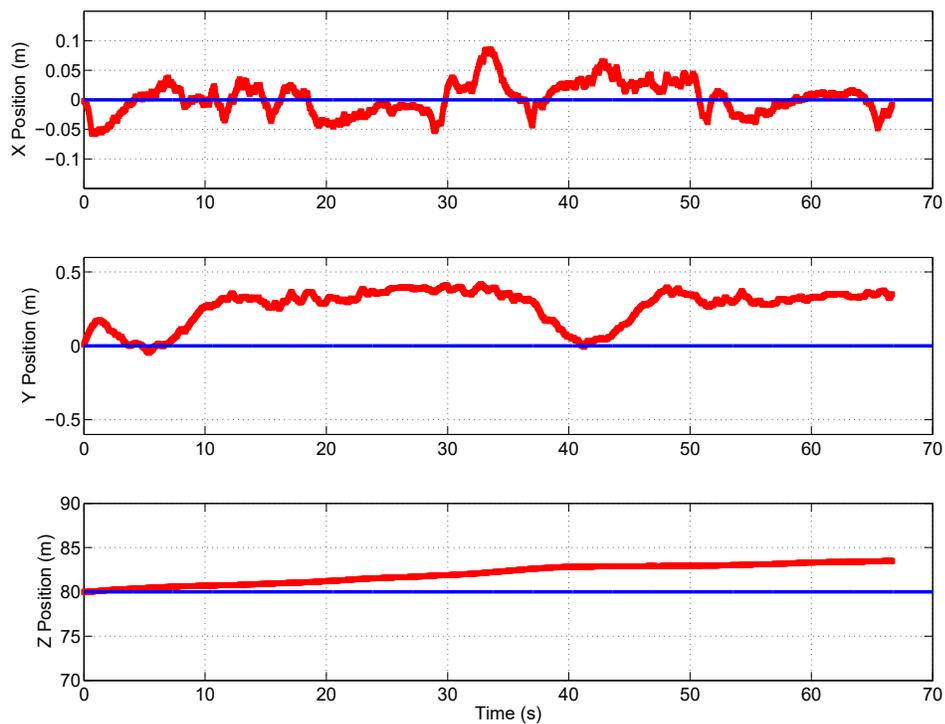


(a) Pose Overlay

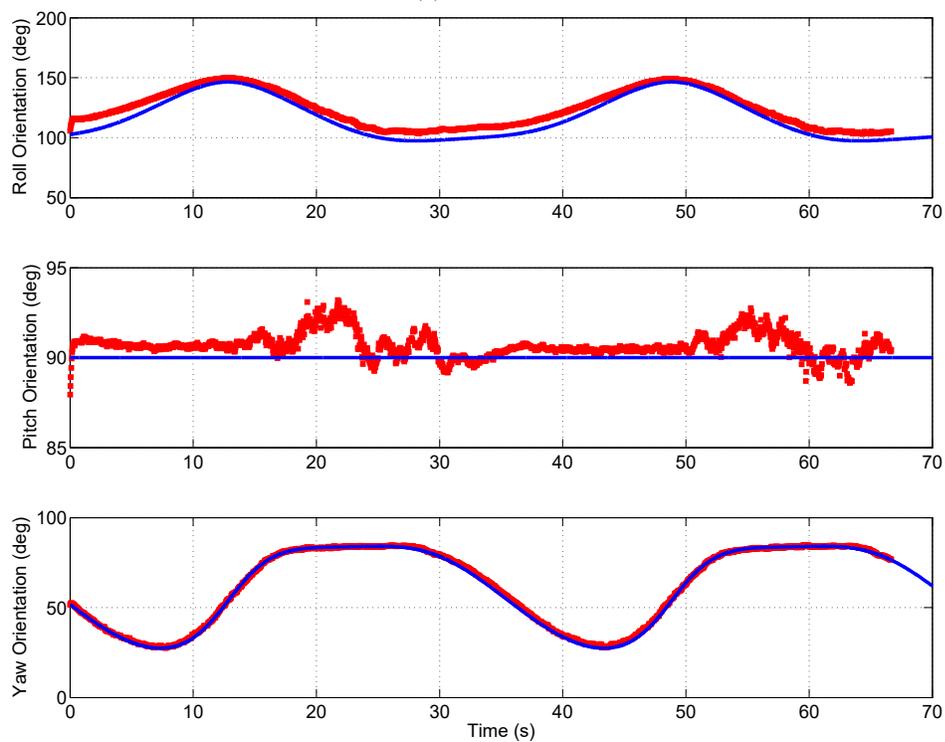


(b) 3D Pose

Figure 7.12: Envisat pose estimation using synthetic monocular camera image.



(a) Position



(b) Orientation

Figure 7.13: Envisat pose estimation using synthetic monocular camera images. Red lines are the computed pose, blue lines are the GT.

the handoff between the posterior mask to prior mask moving from frame to frame may accumulate errors with progressively worsening of the prior mask. Based on the results of this experiment, we observe the level-set approach is highly sensitive to the visible region. Interestingly, the results show pose estimation can be performed to some degree of accuracy even using a highly coarse internal model. Given most satellite bodies have similar shapes and rough dimensional envelopes, the region-based method can use generic internal elements to build a spacecraft model and eliminate the requirements to have precise target geometry, which could lead to a semi-supervised or unsupervised approach.

7.3.6 ISS Pose Estimation

Figure 7.15 provides the region-based pose estimation results for the STS-135 mission during the ISS docking and undocking phase. In Fig. 7.15, columns 1 and 2 are from the docking phase, columns 3 and 4 are from the undocking phase. The first image in Fig. 7.15 is the initial pose misalignment with the template histogram underneath, which is computed using a prior mask. The red line in Fig. 7.15 represents the foreground, and the blue line represents the background intensities. The histograms have 256 bins to represent each pixel intensity. The first and last two bins are omitted to prevent extreme intensity counts from overwhelming the rest of the values. For clear illustration, the foreground and background histogram values are normalised. The second image in Fig. 7.15 is the converged pose estimation with the associated foreground and background histograms underneath. The sample image frames contain the initial pose misalignment and the final converged solution. The template histogram and the histograms for each iteration step are below the projection silhouette images. The region estimation uses a high-resolution ISS internal model. Initial translation and rotation misalignments are 8.7 meters and 8.7 degrees RSS respectively. The simplified internal 3D model omits the solar panel truss sections. The radiation panel in the 3D model has a 90 degrees offset from the flight configuration. Despite the differences in the internal model and the actual ISS flight configuration, the region-based method was still able to converge to the correct pose.

Figure 7.16 provides the 6-DOF motion sequence of the SSO relative to the ISS during the docking phase of the STS-135 flight mission. In Fig. 7.16, the top 2 rows provide flight images of the ISS captured by the SSO *TriDAR* thermal camera with internal model mesh

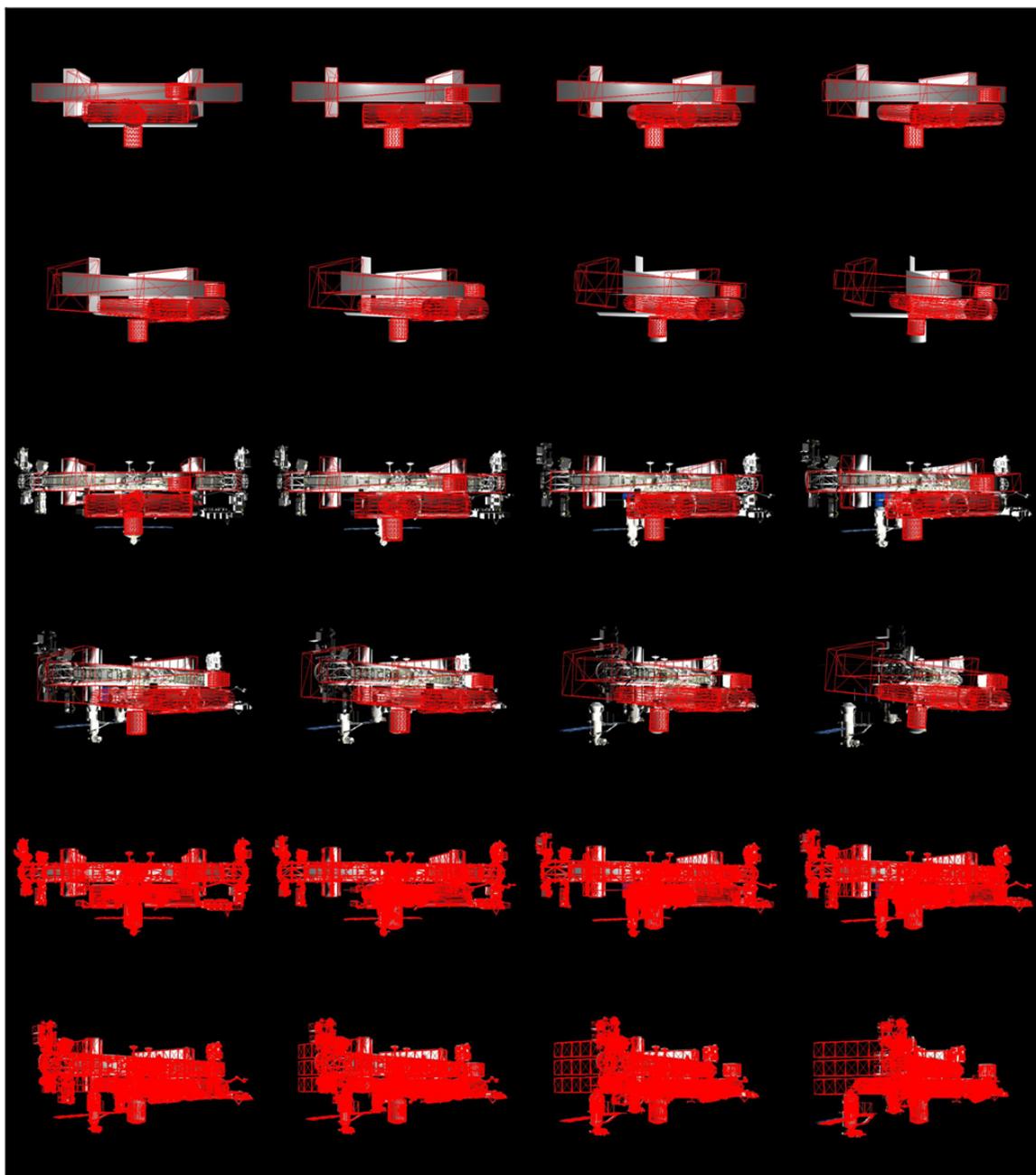


Figure 7.14: ISS CAD model synthetic image pose estimation.

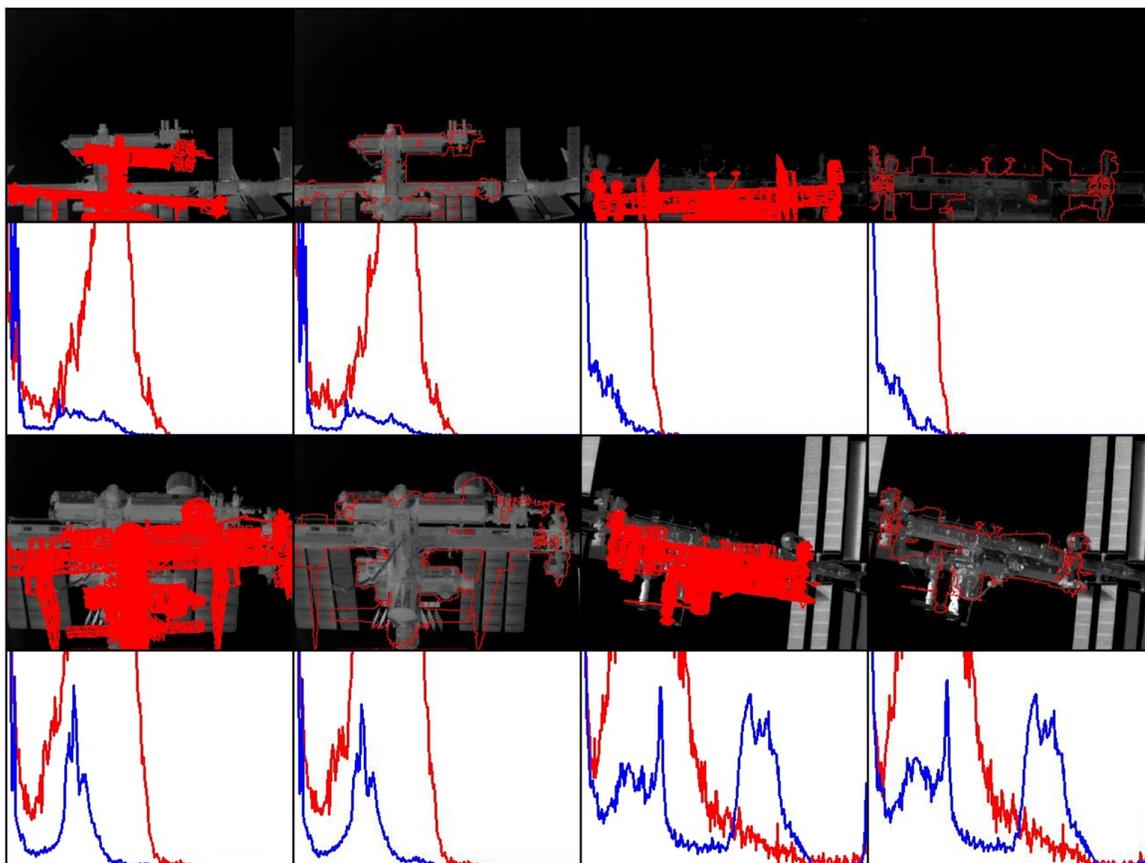


Figure 7.15: STS-135 ISS docking and undocking phase pose estimation using the region-based method.

projection overlay; the bottom 2 rows provide the corresponding 3D model reconstruction of the motion sequence based on the computed pose estimation. The pose projection overlay on the infrared image is in red. The 3D vehicle motion reconstruction is below the projection overlay images. The results show tracking of the ISS image can be generally performed using the monocular camera region-based method; however, after tracking roughly 10 images, a bias begins to accumulate, and reinitialisation is needed to maintain precise pose tracking. The region-based pose estimation method is highly dependent on the quality of the object's visible region. Factors causing higher errors include: close views of the target where the region boundary exceeds the image or covers a majority of the image, shadowing that distorts the object region, ambiguities in the region boundary, and 3D projection resulting in a small outer perimeter of the target vehicle. It is essential to have a good initial estimate pose; large initial misalignments significantly reduce the chances of convergence to the global minimum. A good initial pose estimate can be achieved by taking high-frequency samples of the video image which will reduce the travel distance of the gradient descent process in reaching the global minimum.

The NASA ISS local orbital coordinate system or the Local-Vertical-Local-Horizontal (LVLH) [429] is defined as the spacecraft orbital coordinate that is originated at the vehicle centre of mass, with X - Z plane in the instantaneous orbit plane at the time of interest. Z points toward the Earth centre, Y is normal to the orbit plane opposite of the orbit momentum, and X completes the right-hand. Figure 7.17 provides the 6-DOF motion sequence of the SSO undocking from the ISS initially departing along the *forward* V-bar (LVLH X axis), then turns to the negative H-bar (LVLH Y axis) and performing a proximity flyby *overhead* the negative R-bar (LVLH $-Z$ axis) from ISS *starboard* to *port* [430]. Figure 7.17 shows the STS-135 ISS undocking and proximity flyby sequence thermal image pose estimation*. Figure 7.17(a) is the projection overlay of the internal 3D model on the ISS thermal image. Figure 7.17(b) is the silhouette outline of the projection overlay while using the *fst+* saliency detection. The *fst+* saliency mask is applied to the input thermal image. Figure 7.18 provides the 3D reconstruction of the camera trajectory expressed in the ISS Body Coordinate System (ISSBCS) [429]. Figure 7.19 provides the temporal plots of

*The full estimation sequence video is available at <https://youtu.be/U4xldh-YWos>

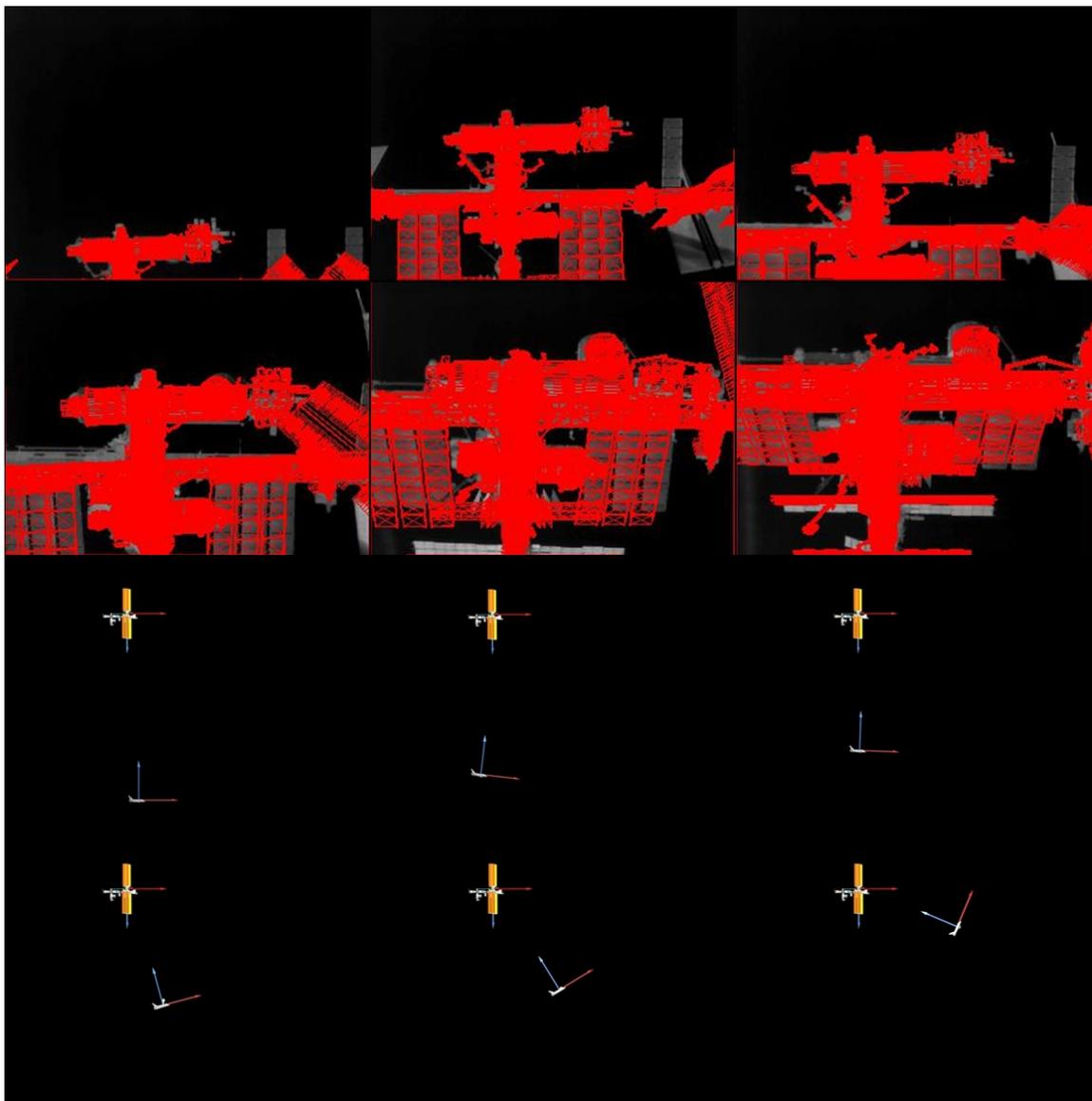


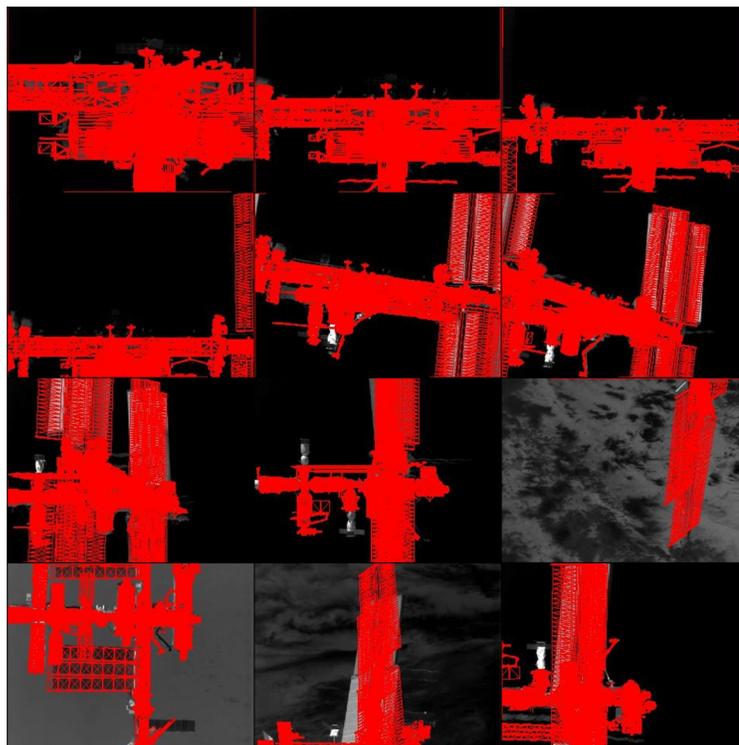
Figure 7.16: STS-135 ISS docking sequence thermal image pose estimation and 3D model pose reconstruction.

the pose estimation for the undocking and flyby sequence, where Figs. 7.19(a) and (b) provides the position and orientation of the camera pose with respect to time respectively. The camera pose is defined as the position from the camera frame to the ISSBCS expressed in the camera frame, and orientation of the ISSBCS rotated from the camera frame using the pitch-yaw-roll Euler angle rotation sequence. The initial condition of each pose estimation frame is the previous frame's pose result.

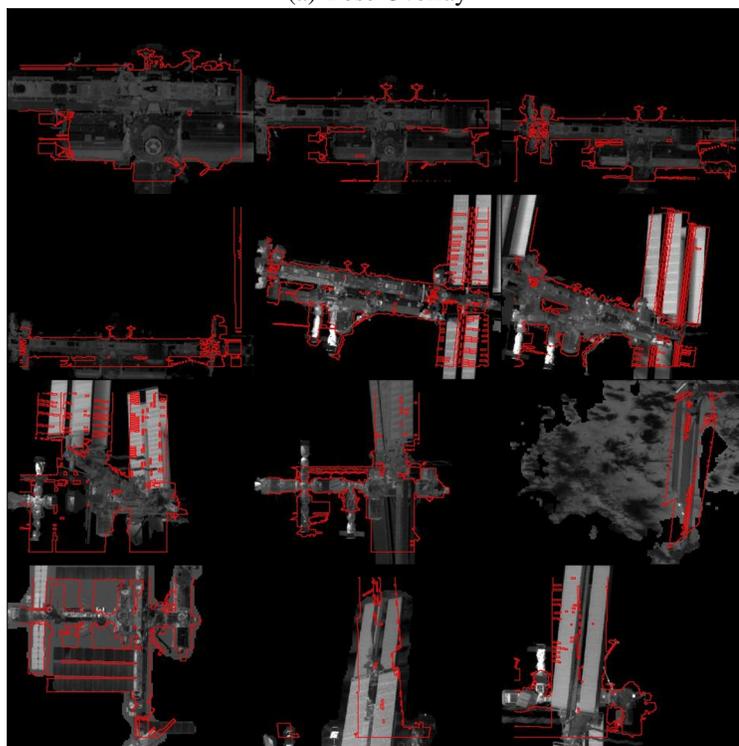
Similar to the docking sequence, the initial condition estimated pose need to be reinitialised every ten frames on average to avoid error build-up. Certain phases of the ProxOps are more robust than others; for example, the initial V-bar departure, the negative R-bar overhead transition can be estimated more precisely than the V-bar to H-bar turn, the turn to R-bar, and turning from R-bar to H-bar. The V-bar departure and R-bar transition are mostly translation movements while the turning sequences require changing the roll and pitch-axis Euler angles as shown in Fig. 7.18. In Fig. 7.18, the camera frame with respect to the ISSBCS is shown in red squares; a to-scale simplified ISS model outline is shown in blue lines; the ISSBCS is located at the ISS mass centre and is in the general direction of the ISS LVLH coordinate system with slight attitude offsets. The difficulty in performing gradient descent using the level-set segmentation comes from two main factors. First, the exact detail of the ISS configuration is not known; for example, the solar panel pan and tilt angles, the SSRMS joint configuration and base location, the radiation panel orientations, the attached Soyuz module location, shape, and their solar panel deployment positions. All of these unknowns adds error to the internal model projection on the thermal image and reduce chances in full alignment of the projection with the captured image. Secondly, throughout the entire motion sequence, the ISS is not viewed in its entirety. The region method performs best when the entire vehicle is displayed in the image with distinctive shape features. Conversely, when the projected region almost entirely covers the image such as frames 1 and 366 in Fig. 7.17 or only a tiny portion of the vehicle is displayed such as frames 585 and 731, the region method performs poorly due to ambiguity in the region silhouette. The sequence around frame 731 is especially difficult to predict since it is the combination of both error factors. The estimated range is comparable to the *TriDAR* measured values.

Based on the experiment results, it is recommended to include additional correction

methods to increase prediction accuracy such as using feature localisation when the target is near, and there are an abundant amount of image features within the boundary region that can be used to clarify silhouette ambiguity. Secondly, the pose estimation accuracy can improve by combining stochastic filters with the camera pose estimation, taking into account predictions in the dynamic motion, namely, implementing a Kalman filter. Local histogram template matching can also reduce region ambiguity as demonstrated by Hexner [283] and Tjaden [284].



(a) Pose Overlay



(b) Projected Silhouette

Figure 7.17: STS-135 ISS undocking and proximity flyby sequence thermal image pose estimation. Sequence frames 1, 74, 147, 220, 293, 366, 439, 512, 585, 658, 731, and 793 are displayed respectively.

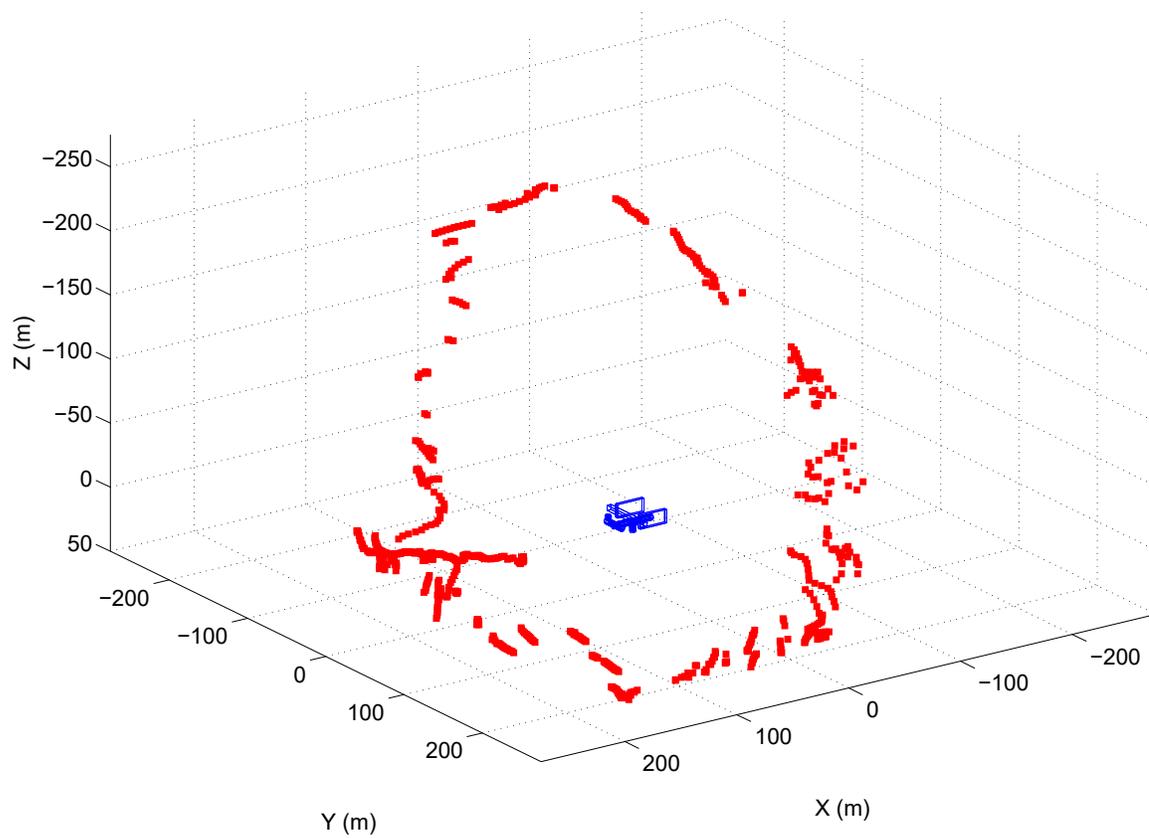
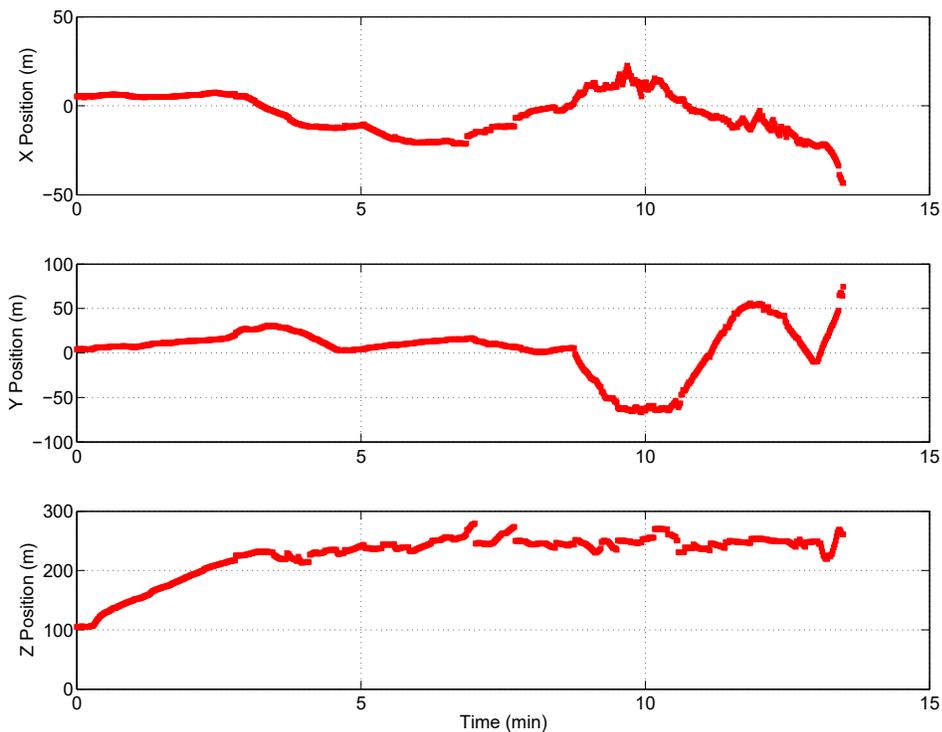
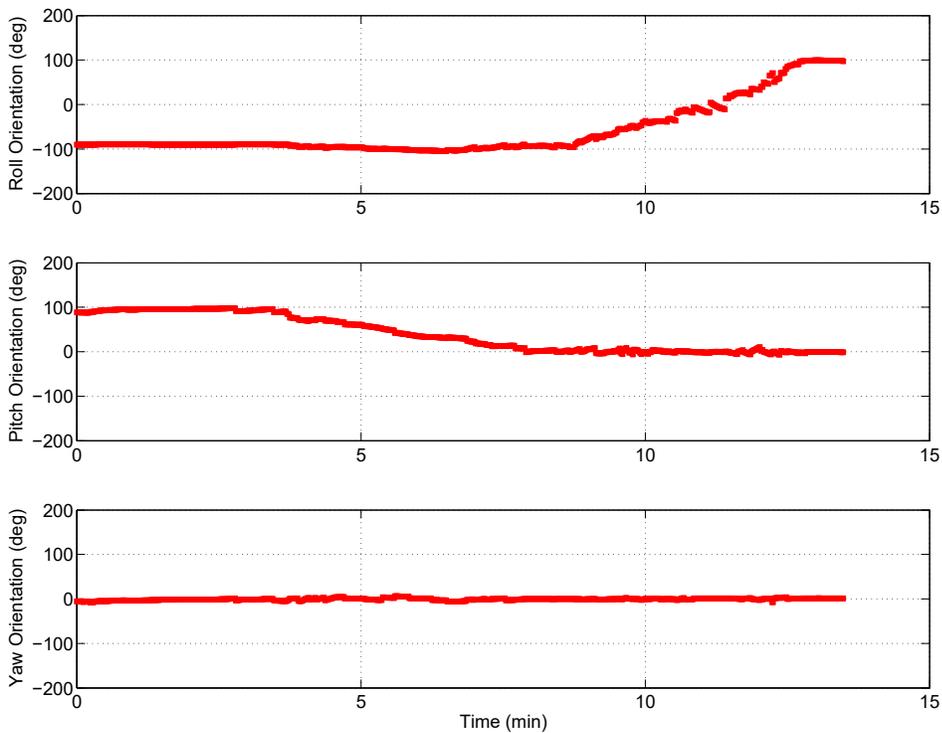


Figure 7.18: STS-135 ISS undocking and proximity flyby sequence 3D reconstruction.



(a) Position



(b) Orientation

Figure 7.19: STS-135 ISS undocking and proximity flyby sequence pose estimation.

Chapter 8

Conclusion

8.1 Summary

In conclusion, we comprehensively investigate monocular camera pose estimation for spacecraft navigation. Our method focuses on image processing, target extraction, and pose estimation using model-based image registration. We use HER, HT and point inflation to build image point map from edges for PnP correspondence. We also compared 12 image key-point and descriptor methods and found SIFT and BRIEF with Harris corner out-perform the other methods in precision and speed respectively. We developed the yBRIEF descriptor that outperforms the BRIEF descriptor; however, handcrafted features can be unreliable when it comes to image viewpoint changes. To simplify the image for pose estimation, we tested bounding box localisation and image segmentation. We tested BoVW method using PCA and five different classifiers and found the texton and k-means approach to be impractical for real-time operations. We also compare the results of the BoVW with ConvNet approaches and found the ConvNet methods strongly outperformed BoVW; however, for some image scenarios such as the ISS infrared video sequence, training the Inception-ResNet and ResNet networks leads to unstable results due to lack of unique training images. In our semantic segmentation effort, we found the CNN methods have similar performance as background subtraction techniques. However, CNN autoencoders require large amounts of labelled image data for front-end training; this was too expensive to implement practically. The best solution for image extraction was using image saliency methods. Image saliency does not require front-end training and can generate relatively stable spacecraft foreground mask in real-time. We developed three methods based on GMR specialising in precision or in speed. We further developed a saliency technique called *fst+* specifically for spacecraft images. Our saliency detection method outperformed the state-of-the-art methods using the SatSeg dataset. We use the *fst+* method to generate front end mask for the

ISS undocking thermal video. We also developed a FC-HSF method for RSM thermal images, where we combine high frequency response edge features with the monochromatic image to enhance the gradient descent landscape. Our methods showed superior performance when combined with the pose estimation scheme. We also created a Earth passage detection algorithm that is effective in the ISS demonstration image sequence.

We compared various pose estimation methods including appearance based PCA method with *ePCA* occlusion optimisation, SoftPOSIT, *ePnP* and region-based pose estimation approach. The appearance-based method is too restrictive and requires large computation resources to compute the SVD when image resolution is large; we conclude the appearance-based method cannot be practically implemented in real-time to handle all camera views robustly. We added RANSAC and a sequence sentinel to improve robustness of using image features and *PnP*. While our enhancements to SoftPOSIT initialisation increased its robustness, there are instances when the simulated annealing process still cannot provide adequate convergence, and the iteration process will take too long to complete. *ePnP* is more robust than SoftPOSIT; however, it is less precise during sequential frame tracking. Our tests show the region-based level-set segmentation method is more precise and robust than the *PnP* method partly due to better invariance property of the projected region over the handcrafted image features. We also improve on the PWP3D region method by introducing CoM initialisation and an improved gradient descent scheme. Our novel enhancements resulted in a more stable pose convergence. Our method produced stable results using the Envisat video sequence and was able to perform pose estimation on the ISS undocking thermal image sequence.

Our final proposed framework is an end-to-end spacecraft pose estimation process using an automated DoG-based Earth background detector with *fst+* spacecraft foreground saliency extraction combined with our enhanced region-based pose estimation. Our foreground extraction approach is less tedious to implement than ConvNet-based localisation and semantic segmentation. Our pose estimation approach is more robust than using point correspondence and more efficient, robust and precise than PCA-based appearance models. As previously mentioned in Sec. 1.2 our approach best operate under full view of the spacecraft body; however, this does not exclude the possibility to use 3D sub-component models for near-range ProxOps such as docking. Alternative solutions such as feature point

correspondence and SfM or SLAM may also be good candidates for near-range ProxOps where image features are abundant.

8.2 Future Work and Recommendations

Our experience shows precise foreground extraction cannot be purely image-driven, future work shall include combining ConvNet methods with saliency detection for precise foreground extraction. Our tests show the region-based pose estimation will result in an error build up over time. We recommend including a Kalman Filter in the future to improve the robustness of the pose prediction. We showed our region-based pose estimation method to also work with low-resolution internal models. We have demonstrated the possibility of using generic shape models in both point-based homography projection and level-set region method; this can lead to solutions to a semi-known problem with little knowledge of the target spacecraft. Future work could further explore increasing prediction precision from using generic shapes instead of complex internal models for faster solution and flexible mission application. More rigorous pose estimation performance analysis may follow industry standards as described by American Society for Testing and Materials (ASTM) E2919-14 Standard Test Method for Evaluating the Performance of Systems that Measure Static, 6-DOF Pose and ASTM E3064-16 Standard Test Method for Evaluating the Performance of Optical Tracking Systems that Measure 6-DOF Pose. When the pose estimation system matures it may be useful to develop generic *expectivity index* and *pose ambiguity metrics* to aid general engineering design [431,432].

Bibliography

- [1] Purves, D., Augustine, G., D., F., Katz, L., LaMantia, A., McNamara, J., and Williams, S., *Neuroscience*, Sinauer Associates, Sunderland, MA, 2nd ed., 2001.
- [2] He, K., Zhang, X., Ren, S., and Sun, J., “Deep residual learning for image recognition,” *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, 2015.
- [3] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S. Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, “Going deeper with convolutions,” *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, 2015, pp. 1–9.
- [4] Szegedy, C., Ioffe, S., and Vanhoucke, V., “Inception-v4, Inception-ResNet and the impact of residual connections on learning,” *arXiv preprint*, arXiv:1602.07261, 2016.
- [5] KaewTraKulPong, P. and Bowden, R., “An improved adaptive background mixture model for real-time tracking with shadow detection,” *Proc. of the European Workshop on Advanced Video Based Surveillance Systems*, 2001.
- [6] Zivkovic, Z., “Improved adaptive gaussian mixture model for background subtraction,” *Proc. of the Intl. Conf. on Pattern Recognition*, 2004.
- [7] Zivkovic, Z. and van der Heijden, F., “Efficient adaptive density estimation per image pixel for the task of background subtraction,” *Pattern Recognition Letters*, Vol. 27, No. 7, 2006, pp. 773–780.
- [8] Godbehere, A., Matsukawa, A., and Goldberg, K., “Visual tracking of human visitors under variable-lighting conditions for a responsive audio art installation,” *American Control Conf.*, 2012, pp. 4305–4312.
- [9] Heikkilä, J. and Silvén, O., “A real-time system for monitoring of cyclists and pedestrians,” *Journal of Image and Vision Computing*, 1999, pp. 563–570.
- [10] Alcantarilla, P., Nuevo, J., and Bartoli, A., “Fast explicit diffusion for accelerated features in nonlinear scale spaces,” *British Machine Vision Conf.*, 2013.
- [11] Alahi, A., Ortiz, R., and Vandergheynst, P., “FREAK: fast retina keypoint,” *IEEE Conf. on Computer Vision and Pattern Recognition*, 2012, pp. 510–517.
- [12] Leutenegger, S., Chli, M., and Siegwart, R., “BRISK: binary robust invariant scalable keypoints,” *IEEE Intl Conf. on Computer Vision*, 2011, pp. 2548–2555.
- [13] Calonder, M., Lepetit, V., Strecha, C., and Fua, P., “BRIEF: binary robust independent elementary features,” *European Conf. on Computer Vision*, 2010.

- [14] Bay, H., Ess, A., Tuytelaars, T., and Gool, L., “Speeded-up robust features,” *Computer Vision and Image Understanding*, Vol. 110, No. 3, Jun 2008, pp. 346–359.
- [15] Lowe, D., “Distinctive image features from scale-invariant keypoints,” *Intl. Journal of Computer Vision*, Vol. 60, No. 2, 2004, pp. 91–110.
- [16] Fehse, W., *Automated rendezvous and docking of spacecraft*, Cambridge University Press, Cambridge, UK, 2003.
- [17] Sotro, E. and Bastante, J., “System and GNC concept for rendezvous into elliptical orbit for Mars sample return mission,” *AIAA Guidance, Navigation and Control Conf. and Exhibit*, AIAA, Hilton Head, South Carolina, Aug 2007.
- [18] ESA, “Europe’s automated ship docks to the ISS,” *European Space Agency News*, 2008.
- [19] Castellani, L., Llorente, S., Fernandez, J.M. Ruiz, M., Mestreau-Garreau, A., Cropp, A., and Santovincenzo, “PROBA-3 mission,” *Intl. Journal of Space Science and Engineering*, Vol. 1, No. 4, 2013, pp. 349–366.
- [20] Fourie, D., Tweddle, B., Ulrich, S., and Saenz-Otero, A., “Flight results of vision-based navigation for autonomous spacecraft inspection of unknown objects,” *AIAA Journal of Spacecraft and Rockets*, Vol. 51, No. 6, Nov-Dec 2014, pp. 2016–2026.
- [21] Miravet, C., Pascual, L., Krouch, E., and delCura, J., “An image-based sensor system for autonomous rendez-vous with uncooperative satellites,” *Intl. ESA Conf. on Guidance, Navigation and Control Systems*, Tralee, Ireland, Jun 2008.
- [22] Khan, F., “Mobile internet from the heavens,” *arXiv:1508.02383*, 2015.
- [23] Henshaw, C., Healy, L., and Roderick, S., “LIIVe: a small, low-cost autonomous inspection vehicle,” *AIAA SPACE Conf. and Exposition*, Reston, VA, 2009.
- [24] Kanani, K., “Vision based navigation for debris removal missions,” *Proc. of the Intl. Astronautical Congress*, Naples, Italy, Oct 2012.
- [25] Geller, D. K., “Orbital rendezvous: when is autonomy required?” *AIAA Journal of Guidance, Control, and Dynamics*, Vol. 30, No. 4, 2007, pp. 974–981.
- [26] Chullen, C. and Blome, E., “H-II transfer vehicle and the operations concept for extravehicular activity hardware,” *Intl. Conf. on Environmental Systems*, AIAA, Portland, Oregon, July 2011.
- [27] SpaceX, “Dragon lab,” <http://www.spacex.com/sites/spacex/files/pdf/DragonLabFactSheet.pdf>, Accessed: Sept 6, 2016.
- [28] NASA, “Russian progress spacecraft,” http://www.nasa.gov/mission_pages/station/structure/elements/progress.html, Accessed: Sept 6, 2016.

- [29] USAF, “Fact sheet: Automated Navigation and Guidance Experiment for Local Space (ANGELS),” Jul 2014, <http://www.kirtland.af.mil/shared/media/document/AFD-131204-039.pdf>, Accessed: March 6, 2016.
- [30] Eremenko, P., “Innovation in the age of the third aerospace revolution,” *AIAA Aviation Forum Keynote Address*, AIAA, Denver, CO, Jun 2017.
- [31] Poghosyan, A. and Golkar, A., “CubeSat evolution: analyzing CubeSat capabilities for conducting science missions,” *Progress in Aerospace Science*, Vol. 88, 2017, pp. 59–83.
- [32] Sarda, K., Eagleson, S., Caillibot, E., Grant, C., Kekez, D., Paranajaya, F., and Zee, R., “Canadian advanced nanospace experiment 2: scientific and technological innovation on a three-kilogram satellite,” *Proc. of the Intl. Astronautical Congress*, Fukuoka, Japan, Oct 2005.
- [33] Forshaw, J., Aglietti, G., Navarathinam, N., and Kadhem, H., “An in-orbit active debris removal mission - REMOVEDEBRIS: pre-launch update,” *Proc. of the Intl. Astronautical Congress*, Jerusalem, Israel, Oct 2015.
- [34] Richard, M., Kronig, L., Belloni, F., Rossi, S., Gass, V. Araomi, S., Gavrilovich, I., Shea, H., Paccolat, C., and Thiran, J., “Uncooperative rendezvous and docking for MicroSats, the case for CleanSpace One,” *Intl. Conf. on Recent Advances in Space Technologies*, Istanbul, Turkey, Jun 2013.
- [35] Sansone, F., Branz, F., and Francesconi, A., “A relative navigation sensor for CubeSats based on retro-reflective markers,” *IEEE Intl. Workshop on Metrology for AeroSpace*, IEEE, Padua, Italy, Jun 2017.
- [36] Polites, M., “Technology of automated rendezvous and capture in space,” *AIAA Journal of Spacecraft and Rockets*, Vol. 36, No. 2, Mar-Apr 1999, pp. 280–291.
- [37] Kawano, I., Mokuno, M., Kasai, T., and Suzuki, T., “Result of autonomous rendezvous docking experiment of Engineering Test Satellite-VII,” *AIAA Journal of Spacecraft and Rockets*, Vol. 38, No. 1, 2001, pp. 105–111.
- [38] Hintze, G., Cornett, K., Rahmatipour, M., and A.F., H., “AVGS, AR&D for satellites, ISS, the Moon, Mars and beyond,” *AIAA Conf. and Exhibit*, Rohnert Park, CA, May 2007, AIAA 2007-2883.
- [39] Leinz, M. R., Chen, C. T., Scott, P., Gaumer, W. B., Sabasteanski, P. W., and Beaven, M., “Modeling, simulation, testing, and verification of the orbital express Autonomous Rendezvous and Capture Sensor System (ARCSS),” *Proc. of SPIE Sensors and Systems for Space Applications II*, Vol. 6958, Bellingham WA, 2008, 6958-0C.

- [40] Howard, R., Bryan, T., Lee, J., and Robertson, B., "Next generation advanced video guidance sensor development and test," *AAS Guidance and Control Conf.*, Breckenridge, CO, Jan 2009, AAS 09-064.
- [41] LeCroy, J., Hallmark, D., Scott, P., and Howard, R., "Comparison of navigation solutions for autonomous spacecraft from multiple sensor systems," *Proc. of SPIE Sensors and Systems for Space Applications II*, Vol. 6958, 2008, 6958-0D.
- [42] ESA, "Rendezvous and docking technology," *ATV Information Kit*, 2008.
- [43] Mills, I., "SVS briefing," *NASA SSP DX22 Presentation*, Houston, Tx, Mar 1999.
- [44] Samson, C., English, C., Deslauriers, A., Christie, I., and Blais, F., "Image and tracking elements of the international space station using a 3D auto-synchronized scanner," *Aerospace/Defence Sensing, Simulation, and Controls*, Orlando, FL, Apr 2002.
- [45] Nister, D., Naroditsky, O., and J., B., "Visual odometry," *IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, 652–659, 2004.
- [46] Scaramuzza, D. and Fraundorfer, F., "Visual odometry, part I: the first 30 years and fundamentals," *IEEE Robotics and Automation Magazine*, dec 2011, pp. 80–92.
- [47] Fraundorfer, F. and Scaramuzza, D., "Visual odometry, part II: matching, robustness, optimization, and applications," *IEEE Robotics and Automation Magazine*, Vol. Jun, 2012, pp. 78–90.
- [48] Bošnjak, M., Matko, D., and Blažič, S., "Quadrocopter hovering using position-estimation information from inertial sensors and a high-delay video system," *IEEE/RSJ Intl. Journal of Intelligent Robot and Systems*, Vol. 67, 2012, pp. 43–60.
- [49] Tournier, G., Valenti, M., and How, J., "Estimation and control of a quadrotor vehicle using monocular vision and Moiré patterns," *AIAA Guidance, Navigation, and Control Conf. and Exhibit*, Keystone, CO, Aug 2006.
- [50] Shiu, Y. and Ahmad, S., "3D location of circular and spherical features by monocular model-based vision," *Conf. Proc. Systems, Man and Cybernetics*, Nov 1989, pp. 576–581.
- [51] Guru, D., Shekar, B., and Nagabhushan, P., "A simple and robust line detection algorithm based on small eigenvalue analysis," *Pattern Recognition Letters*, Vol. 25, No. 1, 2004, pp. 1–13.
- [52] Zheng, Y., Ma, W., and Liu, Y., "Another way of looking at monocular circle pose estimation," *IEEE Int Conf. on Image Processing*, San Diego, CA, Oct 2008.

- [53] Lu, J., Shi, Y., and Wu, S., “Monocular vision-based sensor for autonomous mobile robot localization by circular markers,” *Przegląd Elektrotechniczny*, Vol. 89, No. 1b, 2013, pp. 131–133.
- [54] Ogilvie, A., Allport, J., Hannah, M., and Lymer, J., “Autonomous robotic operations for on-orbit satellite servicing,” *Proc. of SPIE Sensors and Systems for Space Applications II*, Vol. 6958, Orlando, FL, Apr 2008.
- [55] Fitzgibbon, A., Pilu, M., and R.B., F., “Direct least square fitting of ellipses,” *IEEE Trans. on Pattern Analysis Machine Intelligence*, Vol. 21, No. 5, 1999, pp. 476–480.
- [56] Tanaka, H. and Sumi, Y., “A visual marker for precise pose estimation based on microlens array,” *Intl. Conf. on Pattern Recognition*, Tsukuba, Japan, Nov 2012.
- [57] Fang, B., Du, Y., Wu, D., and Wang, C., “Robust vision system for space teleoperation ground verification platform,” *Proc. of Chinese Control Conf.*, Xi’an, China, Jul 2013.
- [58] Tweddle, B. and Saenz-Otero, A., “Relative computer vision-based navigation for small inspection spacecraft,” *AIAA Journal of Guidance, Control, and Dynamics*, Vol. 38, No. 5, 2015, pp. 969–977.
- [59] Miller, D., *Development of resource-constrained sensors and actuators for in-space satellite docking and servicing*, Master’s thesis, Dept. of Aeronautics and Astronautics Engineering, Massachusetts Institute of Technology, Cambridge, MA, 2015.
- [60] Tarabini, L., Gil, J., Gandia, F., Molina, M., delCura, J., and Ortega, G., “Ground guided CX-OLEV rendez-vous with uncooperative geostationary satellite,” *Acta Astronautica*, Vol. 61, Mar 2007.
- [61] Earl, M. and Wade, G., “Observation and analysis of the apparent spin period variations of inactive box-wing geosynchronous resident space objects,” *Proc. of the Intl. Astronautical Congress*, Toronto, ON, 2014.
- [62] Cognion, R., Albuja, A. A., and Scheeres, D. J., “Tumbling rates of inactive GEO satellites,” *Proc. of the Intl. Astronautical Congress*, Toronto, ON, 2014.
- [63] Kucharski, D., Kirchner, G., Koidl, F., Fan C., Carman, R., Moore, C., Dmytrotsa, A., Ploer, M., Bianco, G., Medvedskij, M., Makeyev, A., Appleby, G., Suzuki, M., Torre, J., Zhang, Z., Grunwaldt, L., and Qu, F., “Attitude and spin period of space debris Envisat measured by satellite laser ranging,” *IEEE Trans. on Geoscience and Remote Sensing*, Vol. 52, No. 12, 2014, pp. 7651–7657.
- [64] Allen, A., Langley, C., Mukherji, R., Nimelman, M., de Lafontaine, J., Neveu, D., and Tripp, J., “Full-scale testing and platform stabilization of a scanning LIDAR system for planetary landing,” *Proc. of SPIE Space Exploration Technologies*, Vol. 6960, 2008, 6960-04.

- [65] Allen, A., Langley, C., Mukherji, R., Taylor, A., and Barfoot, T., “Rendezvous LIDAR sensor system for terminal rendezvous, capture, and berthing to the international space station,” *Proc. of SPIE Sensors and Systems for Space Applications II*, Vol. 6958, 2008, 6958-0S.
- [66] Ruel, S., Luu, T., Anctil, M., and Gagnon, S., “Target localization from 3D data for on-orbit autonomous rendezvous & docking,” *IEEE Aerospace Conf.*, Big Sky, MT, Mar 2008.
- [67] Ruel, S., Luu, T., and Berube, A., “Space shuttle testing of the TriDAR 3D rendezvous and docking sensor,” *Journal of Field Robotics*, Vol. 29, No. 4, 2012, pp. 535–553.
- [68] Creamer, N., Hartley, R., Obermark, J., Henshaw, C., Roderick, S., and Hope, J., “Autonomous release of a snagged solar array: technologies and laboratory demonstrations,” *AIAA Journal of Spacecraft and Rockets*, Vol. 51, No. 1, 2014, pp. 86–95.
- [69] Besl, P. and McKay, N., “A method for registration of 3D shapes,” *IEEE Trans. on Pattern Analysis Machine Intelligence*, Vol. 14, No. 2, 1992, pp. 239–256.
- [70] Choudhuri, A., *Target design for lidar-based ICP pose estimation for space vision tasks*, Master’s thesis, Dept. of Aerospace Engineering, Ryerson Univ., Toronto, ON, 2009.
- [71] Lang, S. and Jäger, K., “3D scene reconstruction from IR image sequences for image based navigation update and target detection of an autonomous airborne system,” *SPIE Defense and Security Symp.*, Orlando, FL, 2008.
- [72] Ezra, E., Sharir, M., and Efrat, A., “On the performance of the ICP algorithm,” *Computational Geometry*, Vol. 41, No. 1, 2008, pp. 77–93.
- [73] Barrois, B., Hristova, S., Wöhler, C., Kummert, F., and Hermes, C., *3D pose estimation of vehicle using a stereo camera*, Encyclopedia of Sustainability Science and Technology, Springer New York, 2012.
- [74] Hirschmüller, H., Innocent, P., and Garibaldi, J., “Fast, unconstrained camera motion estimation from stereo without tracking and robust statistics,” *Intl. Conf. on Control, Automation, Robotics and Vision*, Singapore, Dec 2002.
- [75] Tweddle, B., *Computer vision-based localization and mapping of an unknown, uncooperative and spinning target for spacecraft proximity operations*, Ph.D. thesis, Dept. of Aeronautics and Astronautics Engineering, Massachusetts Institute of Technology, Cambridge, MA, 2013.
- [76] Tweddle, B., Saenz-Otero, A., and Miller, D., “Design and development of a visual navigation testbed for spacecraft proximity operations,” *AIAA SPACE Conf. and Exposition*, Reston, VA, 2009.

- [77] Fourie, D., Tweddle, B., Ulrich, S., and Saenz-Otero, A., “Vision-based relative navigation and control for autonomous spacecraft inspection of an unknown object,” *AIAA Guidance, Navigation, and Control Conf.*, Reston, VA, 2013.
- [78] Tweddle, B., Setterfield, T., A., S.-O., Miller, D., and Leonard, J., “Experimental evaluation of onboard, visual mapping of an object spinning in micro-gravity aboard the international space station,” *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems*, Chicago, IL, Sept 2014.
- [79] Bridges, C., Taylor, B., Horri, N., Underwood, C., Kenyon, S., Barrera-Ars, J., Pryce, L., and Bird, R., “STRaND-2: visual inspection, proximity operations & nanosatellite docking,” *IEEE Aerospace Conf.*, Big Sky, MT, Mar 2013.
- [80] Pizer, S., Johnston, R., Ericksen, J., Yankaskas, B., and Muller, K., “Contrast-limited adaptive histogram equalization speed and effectiveness,” *Proc. of Conf. on Visualization in Biomedical Computing*, 1990, pp. 337–345.
- [81] Liwicki, S., Tzimiropoulos, G., Zafeiriou, S., and Pantic, M., “Euler principal component analysis,” *Intl. Journal of Computer Vision*, Vol. 101, No. 3, 2013, pp. 498–518.
- [82] Shi, J., Ulrich, S., and Ruel, S., “Spacecraft pose estimation using principal component analysis and a monocular camera,” *Proc. of the AIAA Guidance, Navigation, and Controls Conf. and Exhibit*, Grapevine, TX., Jan 2017.
- [83] Duda, R. and Hart, P., “Use of the Hough transformation to detect lines and curves in pictures,” *Communications of the ACM*, Vol. 15, No. 1, 1972, pp. 11–15.
- [84] Itti, L., Koch, C., and Niebur, E., “A model of saliency-based visual attention, for rapid scene analysis,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 20, No. 11, 1998, pp. 1254–1259.
- [85] McIvor, A., “Background subtraction techniques,” *Proc. of Image and Vision Computing*, Vol. 4, 2000, pp. 3099–3104.
- [86] Krizhevsky, A., Sutskever, I., and Hinton, G., “Imagenet classification with deep convolutional neural networks,” *Neural Information Processing Systems*, 2012.
- [87] Ren, S., He, K., Girshick, R., and Sun, J., “Faster R-CNN: towards real-time object detection with region proposal networks,” *Neural Information Processing Systems*, 2015.
- [88] Badrinarayanan, V., Handa, A., and Cipolla, R., “Segnet: a deep convolutional encoder-decoder architecture for robust semantic pixel-wise labelling,” *Computing Research Repository*, Vol. abs/1505.07293, 2015.

- [89] Rosenhahn, B., *Pose estimation revisited*, Technical Report 0308, Christian Albrechts-Universität zu Kiel Institut für Informatik und Praktische Mathematik, 2003.
- [90] Rosenhahn, B., Perwass, C., and Sommer, G., “Foundations about 2D-3D pose estimation,” *CVonline: The Evolving Distributed, Non-Proprietary, On-line Compendium of Computer Vision*, Edinburgh, 2004.
- [91] Pizzoli, M., Forster, C., and Scaramuzza, D., “REMODE: probabilistic, monocular dense reconstruction in real time,” *IEEE Intl Conf. on Robotics and Automation*, Hong Kong, China, May 2014.
- [92] Doignon, C., “An introduction to model-based pose estimation and 3D tracking technique,” *Scene Reconstruction Pose Estimation and Tracking*, Bellingham WA, 2007, Rustam Stolkin (Ed.).
- [93] Kalman, R., “A new approach to linear filtering and prediction problems,” *Journal of Basic Engineering*, Vol. 82, No. 1, 1960, pp. 35–45.
- [94] Fischler, M. and Bolles, R., “Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography,” *Graphics and Image Processing*, Vol. 24, No. 6, 1981, pp. 381–395.
- [95] Sim, T., Hong, G., and K.B., L., “Modified simith predictor with Dementhon-Horaud pose estimation algorithm for 3D dynamic visual servoing,” *Robotica*, Vol. 20, No. 6, 2002, pp. 615–624.
- [96] Aghili, F. and Parsa, K., “Motion and parameter estimation of space objects using laser-vision data,” *AIAA Journal of Guidance, Control, and Dynamics*, Vol. 32, No. 2, 2009, pp. 537–549.
- [97] Assa, A. and Janabi-Sharifi, F., “A robust vision-based sensor fusion approach for real-time pose estimation,” *IEEE Trans. on Cybernetics*, Vol. 44, No. 2, 2014, pp. 217–227.
- [98] Cremers, D., Kohlberger, T., and Schnörr, C., “Nonlinear shape statistics in Mumford-Shah based segmentation,” *European Conf. on Computer Vision*, 2002, pp. 516–518.
- [99] Yang, C., Zhang, L., Lu, H., Ruan, X., and Yang, M., “Saliency detection via graph-based manifold ranking,” *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, 2013, pp. 3166–3173.
- [100] David, P., Dementhon, D., Duraiswami, R., and Samet, H., “SoftPOSIT: simultaneous pose and correspondence determination,” *Intl. Journal of Computer Vision*, Vol. 59, No. 3, 2004, pp. 259–289.

- [101] Prisacariu, V. and Reid, I., “PWP3D: real-time segmentation and tracking of 3D objects,” *Intl. Journal of Computer Vision*, Vol. 98, No. 3, 2012, pp. 335–354.
- [102] Jasiobedski, P., Greenspan, M., and Roth, G., “Pose determination and tracking for autonomous satellite capture,” *Proc. of Intl. Symp. on Artificial Intelligence and Robotics and Automation in Space*, St-Hubert, QC, Jun 2001.
- [103] Weber, M. and Perona, P., “Unsupervised learning of models for recognition,” *European Conf. on Computer Vision*, Dublin, Ireland, June 2000, pp. 18–32.
- [104] Mikolajczyk, K., Leibe, B., and Schiele, B., “Multiple object class detection with a generative model,” *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, Vol. 1, New York, NY, June 2006, pp. 26–36.
- [105] Ferrari, V., Jurie, F., and Schmid, C., “From images to shape models for object detection,” *Intl. Journal of Computer Vision*, Vol. 87, No. 3, 2010, pp. 284–303.
- [106] Drauschke, M. and Mayer, H., “Evaluation of texture energies for classification of facade images,” *Proc. of Congress of ISPRS*, 2010, pp. 257–262.
- [107] Papageorgiou, C., Oren, M., and Poggio, T., “A general framework for object detection,” *IEEE Intl. Conf. on Computer Vision*, 1998, pp. 555–562.
- [108] Viola, P. and Jones, M., “Robust real-time face detection,” *Intl. Journal of Computer Vision*, Vol. 57, No. 2, 2004, pp. 137–154.
- [109] Csurka, G., Dance, C., Fan, L., Willamowski, J., and Bray, C., “Visual categorization with Bags of Keypoints,” *European Conf. on Computer Vision*, Vol. 1, 2004.
- [110] Shi, J., Ulrich, S., and Ruel, S., “Spacecraft component recognition using a codebook of texton images,” *Proc. of the AIAA Space Conf. and Exhibit*, Orlando, FL, Sept 2017.
- [111] Malik, J. and Perona, P., “Preattentive texture discrimination with early vision mechanisms,” *Journal of the Optical Society of America A*, Vol. 7, No. 5, 1990, pp. 923–932.
- [112] Julesz, B., “Textons, the elements of texture perception, and their interactions,” *Nature*, Vol. 290, No. 5802, 1981, pp. 91–97.
- [113] Malik, J., Belongie, S., Leung, T., and Shi, J., “Contour and texture analysis for image segmentation,” *Intl. Journal of Computer Vision*, Vol. 43, No. 1, 2001, pp. 7–27.
- [114] Cula, O. and Dana, K., “3D texture recognition using bidirectional feature histograms,” *Intl. Journal of Computer Vision*, Vol. 59, No. 1, 2004, pp. 34–60.

- [115] Varma, M. and Zisserman, A., “A statistical approach to texture classification from single images,” *Intl. Journal of Computer Vision*, Vol. 62, No. 1, 2005, pp. 61–81.
- [116] Leung, T. and Malik, J., “Representing and recognizing the visual appearance of materials using three-dimensional textons,” *Intl. Journal of Computer Vision*, Vol. 43, No. 1, 2001, pp. 29–44.
- [117] Schmid, C., “Constructing models for content-based image retrieval,” *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, Vol. 2, Kauai, Hawaii, Dec 2001, pp. 39–45.
- [118] Fei-Fei, L. and Perona, P., “A Bayesian hierarchical model for learning natural scene categories,” *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, Vol. 2, 2005, pp. 524–531.
- [119] Dalal, N. and Triggs, B., “Histograms of oriented gradients for human detection,” *IEEE Conf. on Computer Vision and Pattern Recognition*, San Diego, CA, Jun 2005.
- [120] Felzenszwalb, P., Girshick, R., McAllester, D., and Ramanan, D., “Object detection with discriminatively trained part based models,” *IEEE Trans. on Pattern Analysis Machine Intelligence*, Vol. 32, No. 9, 2010, pp. 1627–1645.
- [121] Vapnik, V., *The nature of statistical learning theory*, Springer, 1995.
- [122] Zhang, H. and Jiang, Z., “Multi-view space object recognition and pose estimation based on kernel regression,” *Chinese Journal of Aeronautics*, Vol. 27, No. 5, 2014, pp. 1233–1241.
- [123] Girshick, R., Iandola, F., Darrell, T., and Malik, J., “Deformable part models are convolutional neural networks,” *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, 2015, pp. 437–446.
- [124] Oquab, M., Bottou, L., Laptev, I., and Sivic, J., “Is object localization for free? Weakly-supervised learning with convolutional neural networks,” *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, 2015.
- [125] Deng, J., Dong, W., Socher, R., Li, L., Li, K., and Fei-Fei, L., “Imagenet: a large-scale hierarchical image database,” *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, 2009.
- [126] Zeiler, M. and Fergus, R., “Visualizing and understanding convolutional networks,” *arXiv preprint*, arXiv:1311.2901, 2013.
- [127] Oquab, M., Bottou, L., Laptev, I., and Sivic, J., “Learning and transferring mid-level image representations using CNN,” *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, 2014.

- [128] Simonyan, K. and Zisserman, A., “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint*, arXiv:1409.1556, 2014.
- [129] He, K., Zhang, X., Ren, S., and Sun, J., “Identity mappings in deep residual networks,” *arXiv preprint*, arXiv:1603.05027, 2016.
- [130] Zagoruyko, S. and Komodakis, N., “Wide residual networks,” *British Machine Vision Conf.*, 2017.
- [131] Girshick, R., Donahue, J., Darrel, T., and Malik, J., “Rich feature hierarchies for accurate object detection and semantic segmentation,” *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, 2014.
- [132] Girshick, R., “Fast R-CNN,” *IEEE Intl. Conf. on Computer Vision*, 2015.
- [133] Liu, W., Anguelov, D., E. C. S. C., and Reed, S., “SSD: single shot multibox detector,” *Computing Research Repository*, Vol. abs/1512.02325, 2015.
- [134] Redmon, J., Divvala, S., Girshick, R., and Farhadi, A., “You only look once: unified, real-time object detection,” *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, 2016.
- [135] Redmon, J. and Farhadi, A., “YOLO9000: Better, faster stronger,” *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, 2017, pp. 6517–6525.
- [136] Howard, A., Zhu, M., Chen, B., Kalenichenko, D. Wang, W., Weyand, T., Andreetto, M., and Adam, H., “Mobilenets: efficient convolutional neural networks for mobile vision applications,” *arXiv:1704.04861*, 2017.
- [137] Huang, J., Rathod, V., Sun, C., Zhu, M., Korattikara, A., Fathi, A., Fischer, I., Wojna, Z., Song, Y., Guadarrama, S., and Murphy, K., “Speed/accuracy trade-offs for modern convolutional object detectors,” *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, Vol. 4, 2017.
- [138] Tripathy, D. and Guru Raghavendra Reddy, K., “Adaptive threshold background subtraction for detecting moving object on conveyor belt,” *Intl. Journal of Indestructible Mathematics and Computing*, Vol. 1, No. 1, 2017, pp. 41–46.
- [139] Otsu, N., “A threshold selection method from gray-level histograms,” *Automatica*, 1975.
- [140] Arbeláez, P., Maire, M., Fowlkes, C., and J., M., “Contour detection and hierarchical image segmentation,” *IEEE Trans. on Pattern Analysis Machine Intelligence*, Vol. 33, No. 5, 2010, pp. 898–916.
- [141] Belongie, S., Carson, C., Greenspan, H., and J., M., “Color- and texture-based image segmentation using EM and its application to content-based image retrieval,” *Intl. Conf. on Computer Vision*, 1998, pp. 675–682.

- [142] Shotton, J., Winn, J., Rother, C., and Criminisi, A., “TexonBoost for image understanding: multi-class object recognition and segmentation by jointly modeling texture, layout, and context,” *Intl. Journal of Computer Vision*, Vol. 81, No. 1, 2009, pp. 2–23.
- [143] Tsai, Y., Yang, M., and Black, M., “Video segmentation via object flow,” *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, 2016.
- [144] Tu, W., He, S., Yang, Q., and Chien, S., “Real-time salient object detection with a minimum spanning tree,” *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, 2016, pp. 2334–2342.
- [145] Dai, J., He, K., and Sun, J., “Instance-aware semantic segmentation via multi-task network cascades,” *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, 2016, pp. 3150–3158.
- [146] Long, J., Shelhamer, E., and Darrel, T., “Fully convolutional networks for semantic segmentation,” *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, 2015, pp. 3431–3440.
- [147] Ronneberger, O., Fischer, P., and Brox, T., “U-Net: convolutional networks for biomedical image segmentation,” *Intl. Conf. on Medical Image Computing and Computer Assisted Intervention*, 2015, pp. 234–241.
- [148] Noh, H., Hong, S., and Han, B., “Learning deconvolution network for semantic segmentation,” *Intl. Conf. on Computer Vision*, 2015.
- [149] Dosovitskiy, A., Springenberg, J., Tatarchenko, M., and Brox, T., “Learning to generate chairs, tables and cars with convolutional networks,” *IEEE Trans. on Pattern Analysis Machine Intelligence*, 2016.
- [150] Yang, J., Price, B., Cohen, S., Lee, H., and Yang, M., “Object contour detection with a fully convolutional encoder-decoder network,” *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, 2016.
- [151] Shelhamer, E., Long, J., and Darrell, T., “Fully convolutional networks for semantic segmentation,” *IEEE Trans. on Pattern Analysis Machine Intelligence*, Vol. 39, No. 4, 2017, pp. 640–651.
- [152] Paszke, A., Chaurasia, A., Kim, S., and Culurciello, E., “ENet: a deep neural network architecture for real-time semantic segmentation,” *arXiv preprint*, arXiv:1606.02147v1, 2016.
- [153] He, K., Gkioxari, G., Dollár, P., and Girshick, R., “Mask R-CNN,” *IEEE Intl Conf. on Computer Vision*, 2017, pp. 2980–2988.

- [154] Isola, P., Zhu, J., Zhou, T., and Efros, A., “Image-to-image translation with conditional adversarial networks,” *arXiv preprint*, arXiv:1611.07004, 2016.
- [155] Borji, A., Cheng, M., Jiang, H., and Li, J., “Salient object detection: a benchmark,” *IEEE Trans. on Image Processing*, Vol. 24, No. 12, 2015, pp. 5706–5723.
- [156] Aye, H. and Zaw, S., “Salient object based action recognition using histogram of changing edge orientation,” *IEEE Intl. Conf. on Software Engineering Research, Management and Applications*, 2017.
- [157] Assens, M., Giro-i Nieto, X., McGuinness, K., and N.E., O., “SaltiNet: scan-path prediction on 360 degree images using saliency volumes,” *IEEE Intl. Conf. on Computer Vision Workshop*, 2017.
- [158] Fattal, A.-K., Karg, M., Scharfenberger, C., and Adamy, J., “Saliency-guided region proposal network for CNN based object detection,” *IEEE Intl. Conf. on Intelligent Transportation System*, 2017.
- [159] Christopoulos, C., Skodras, A., and Ebrahimi, T., “The JPEG2000 still image coding system: an overview,” *IEEE Trans. on Consumer Elec.*, Vol. 46, No. 4, 2002, pp. 1103–1127.
- [160] Yang, J. and Yang, M., “Top-down visual saliency via joint CRF and dictionary learning,” *IEEE Trans. on Pattern Analysis Machine Intelligence*, Vol. 39, No. 3, 2017, pp. 576–588.
- [161] Borji, A., “Boosting bottom-up and top-down visual features for saliency estimation,” *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, 2012, pp. 438–445.
- [162] Chen, T., Lin, L., Liu, L., Luo, X., and Li, X., “DISC: deep image saliency computing via progressive representation learning,” *IEEE Trans. on Neural Network Learning System*, Vol. 27, No. 6, 2016, pp. 1135–1149.
- [163] Borenstein, E. and Malik, J., “Shape guided object segmentation,” *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, 2006, pp. 969–976.
- [164] Ye, L., Liu, Z., Zhou, X., Shen, L., and Zhang, J., “Saliency detection via similar image retrieval,” *IEEE Signal Processing Letters*, Vol. 23, No. 6, 2016, pp. 838–842.
- [165] Xia, C., Li, J., Chen, X., Zheng, A., and Zhang, Y., “What is and what is not a salient object, learning salient object detector by ensembling linear exemplar regressors,” *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, 2017, pp. 4321–4329.
- [166] Wang, J., Jiang, H., Yuan, Z., Cheng, M., Hu, X., and Zheng, N., “Salient object detection: a discriminative regional feature integration approach,” *Intl. Journal of Computer Vision*, Vol. 123, No. 2, 2017, pp. 251–268.

- [167] Cheng, M., Mitra, N., Huang, X., Torr, P., and Hu, S., “Global contrast based salient region detection,” *IEEE Trans. on Pattern Analysis Machine Intelligence*, Vol. 37, No. 3, 2015, pp. 569–582.
- [168] Liu, T., Sun, J., Zheng, N., Tang, X., and Shum, H., “Learning to detect a salient object,” *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, 2007.
- [169] Liu, T., Yuan, Z., Sun, J., Wang, J., Zheng, N., Tang, X., and Shum, H., “Learning to detect a salient object,” *IEEE Trans. on Pattern Analysis Machine Intelligence*, Vol. 33, No. 2, 2011, pp. 353–367.
- [170] Li, J., Tian, Y., Huang, T., and Gao, W., “Probabilistic multi-task learning for visual saliency estimation in video,” *Intl. Journal of Computer Vision*, Vol. 90, No. 2, 2010, pp. 150–165.
- [171] Li, J., Tian, Y., Chen, X., and Huang, T., “Measuring visual surprise jointly from intrinsic and extrinsic contexts for image saliency estimation,” *Intl. Journal of Computer Vision*, Vol. 120, No. 1, 2016, pp. 44–60.
- [172] Xu, Y., Li, J., Chen, J., Shen, G., and Gao, Y., “A novel approach for visual saliency detection and segmentation based on objectness and top-down attention,” *IEEE Intl. Conf. on Image, Vision and Computing*, 2017, pp. 4321–4329.
- [173] Zhang, D., Han, J., and Zhang, Y., “Supervision by fusion: towards unsupervised learning of deep salient object detector,” *IEEE Intl. Conf. on Computer Vision*, 2017.
- [174] Shigematsu, R., Feng, D., You, S., and Barnes, N., “Learning RGB-D salient object detection using background enclosure, depth contrast, and top-down features,” *IEEE Intl. Conf. on Computer Vision*, 2017.
- [175] Luo, Z., Mishra, A., Achkar, A., Eichel, J. Li, S., and Jodoin, P., “Non-local deep features for salient object detection,” *IEEE Intl. Conf. on Computer Vision*, 2017.
- [176] Hu, P., Shuai, B., Liu, J., and Wang, G., “Deep levelsets for salient object detection,” *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, 2017.
- [177] Li, G., Xie, Y., Lin, L., and Yu, Y., “Instance-level salient object segmentation,” *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, 2017, pp. 247–256.
- [178] Borji, A., Cheng, M., Hou, Q., Jiang, H., and Li, J., “Salient object detection: a survey,” *arXiv preprint*, Vol. 2, arXiv:1411.5878, 2014.
- [179] Koch, C. and Ullman, S., “Shifts in selective visual attention: towards the underlying neural circuitry,” *Human Neurobiology*, Vol. 4, 1985, pp. 219–227.
- [180] Hou, X. and Zhang, L., “Saliency detection: a spectral residual approach,” *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, 2007.

- [181] Guo, C., Ma, Q., and Zhang, L., “Spatio-temporal saliency detection using phase spectrum of quaternion Fourier transform,” *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, 2008.
- [182] Zhai, Y. and Shah, M., “Visual attention detection in video sequences using spatiotemporal cues,” *ACM Intl. Conf. on Multimedia*, 2006.
- [183] Achanta, R., Estrada, F., Wils, P., and Süssstrunk, S., “Salient region detection and segmentation,” *IEEE Intl. Conf. on Computer Vision*, 2008.
- [184] Rahtu, E., Kannala, J., Salo, M., and Heikkilä, J., “Segmenting salient objects from images and videos,” *European Conf. on Computer Vision*, IEEE, 2010, pp. 366–379.
- [185] Zhang, J., Sclaroff, S., Lin, Z., Shen, X., Price, B., and Měch, “Minimum barrier salient object detection at 80 FPS,” *IEEE Intl. Conf. on Computer Vision*, 2015.
- [186] Achanta, R., Hemami, S., Estrada, F., and Süssstrunk, “Frequency-tuned salient region detection,” *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, 2009, pp. 1597–1604.
- [187] Zhao, J., Gao, X., Chen, Y., and Feng, H., “Optical imaging system-based real-time image saliency extraction method,” *Optical Engineering*, Vol. 54, No. 4, 2015.
- [188] Wei, Y., Wen, F., Zhu, W., and Sun, J., “Geodesic saliency using background priors,” *European Conf. on Computer Vision*, 2012, pp. 29–42.
- [189] Achanta, R. and Süssstrunk, S., “Saliency detection using maximum symmetric surround,” *IEEE Int Conf. on Image Processing*, Hong Kong, China, Sept 2010.
- [190] Seo, H. and Milanfar, P., “Static and space-time visual saliency detection by self-resemblance,” *Journal of Vision*, Vol. 9, No. 12, 2009, pp. 15.
- [191] Margolin, R., Tal, A., and Zelnik-Manor, L., “What makes a patch distinct?” *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, 2013, pp. 1139–1146.
- [192] Cheng, M., Warrell, J., Lin, W., Zheng, S., Vineet, V., and Crook, N., “Efficient salient region detection with soft image abstraction,” *IEEE Intl. Conf. on Computer Vision*, 2013, pp. 1529–1536.
- [193] Yacoob, Y. and Davis, L., “Segmentation using meta-texture saliency,” *IEEE Intl. Conf. on Computer Vision*, 2007.
- [194] Chan, K., “Saliency/non-saliency segregation in video sequences using perception-based local ternary pattern features,” *IEEE Intl. Conf. on Machine Vision Applications*, 2017.

- [195] Wang, W., Shen, J., Yang, R., and Porikli, F., “Saliency-aware video object segmentation,” *IEEE Trans. on Pattern Analysis Machine Intelligence*, Vol. 40, No. 1, 2018, pp. 20–33.
- [196] Jung, C., Kim, W., Yoo, S., and Kim, C., “A novel monochromatic cue for detecting regions of visual interest,” *Journal of Image and Vision Computing*, Vol. 32, 2014, pp. 405–413.
- [197] Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., and Süsstrunk, “SLIC superpixels compared to state-of-the-art superpixel methods,” *IEEE Trans. on Pattern Analysis Machine Intelligence*, Vol. 34, No. 11, 2012, pp. 2274–2281.
- [198] Li, Z. and Chen, J., “Superpixel segmentation using linear spectral clustering,” *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, 2015, pp. 1356–1363.
- [199] Van den Bergh, M., Boix, X., Roig, G., and Van Gool, L., “SEEDS: superpixels extracted via energy-driven sampling,” *Intl. Journal of Computer Vision*, Vol. 111, No. 3, 2015, pp. 298–314.
- [200] Tian, Z., Zheng, N., Xue, J., Lan, X., Li, C., and Zhou, G., “Video object segmentation with shape cue based on spatiotemporal superpixel neighbourhood,” *IET Computer Vision*, Vol. 8, No. 1, 2014, pp. 16–25.
- [201] Tan, Z., Wan, L., Feng, W., and Pun, C., “Image co-saliency detection by propagating superpixel affinities,” *IEEE Intl. Conf. on Acoustics, Speech and Signal Processing*, 2013.
- [202] Wang, Z. and Wu, X., “Salient object detection using biogeography-based optimization to combine features,” *Applied Intelligence*, Vol. 45, No. 1, 2016, pp. 1–17.
- [203] Li, S., Zeng, C., Fu, Y., and Liu, S., “Optimizing multi-graph learning based salient object detection,” *Signal Processing: Image Communication*, Vol. 55, 2017, pp. 93–105.
- [204] Li, C., Zhang, B., Zhang, S., and Sheng, H., “Saliency detection with relative location measure in light field image,” *IEEE Intl. Conf. on Image, Vision and Computing*, Chengdu, China, June 2017.
- [205] Ye, R. and Chen, Z., “Universal enhancement of salient object detection,” *IEEE Intl. Conf. on Multimedia and Expo*, July 2017.
- [206] Li, L., Zhou, F., Zheng, Y., and Bai, X., “Saliency detection based on foreground appearance and background-prior,” *Neurocomputing*, Vol. 301, 2018, pp. 46–61.
- [207] Qi, W., Cheng, M., Borji, A., Lu, H., and Bai, L., “SaliencyRank: two-stage manifold ranking for salient object detection,” *Computational Visual Media*, Vol. 1, No. 4, 2015, pp. 309–320.

- [208] Wu, X., Lin, X., Jiang, L., and Zhao, D., “An improved manifold ranking based method for saliency detection,” *IEEE Intl. Conf. on Systems and Informatics*, 2017.
- [209] Morency, L. and Gupta, R., “Robust real-time egomotion from stereo images,” *Proc. of Intl. Conf. on Image Processing*, 2003.
- [210] Sinclair, D., Blake, A., and Murray, D., “Robust estimation of egomotion from normal flow,” *Intl. Journal of Computer Vision*, Vol. 13, No. 1, 1994, pp. 57–69.
- [211] Tomasi, C. and Kanade, T., “Shape and motion from image streams under orthography: a factorization method,” *Intl. Journal of Computer Vision*, Vol. 9, No. 2, 1992, pp. 137–154.
- [212] Augenstein, S. and Rock, S., “Improved frame-to-frame pose tracking during vision-only SLAM/SFM with a tumbling target,” *IEEE Intl. Conf. on Robotics and Automation*, Shanghai, China, May 2011.
- [213] Gennery, D., “Visual tracking of known three-dimensional objects,” *Intl. Journal of Computer Vision*, Vol. 7, No. 3, 1992, pp. 243–270.
- [214] Jurie, F. and Dhome, M., “Real time robust template matching,” *British Machine Vision Conf.*, 2002, pp. 123–131.
- [215] Isard, B., “Condensation: conditional density propagation for visual tracking,” *Intl. Journal of Computer Vision*, Vol. 29, No. 1, 1998, pp. 5–28.
- [216] Lan, X. and Huttenlocher, D., “A unified spatio-temporal articulated model for tracking,” *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, Vol. 1, 2004, pp. 722–729.
- [217] Lepetit, V. and Fua, P., “Monocular model-based 3D tracking of rigid objects: A survey,” *Foundations and Trends in Computer Graphics and Vision*, Vol. 1, No. 1, 2005, pp. 1–89.
- [218] Lingenauber, M., Kriegel, S., Kabecker, M., and Panin, G., “A dataset to support and benchmark computer vision development for close range on-orbit servicing,” *Symp. on Advanced Space Technologies in Robotics and Automation*, Noordwijk, Netherlands, Nov 2015.
- [219] Haralick, R., Chu, Y., Watson, L., and Shapiro, L., “Matching wire frame objects from their 2D perspective projection,” *Pattern Recognition*, Vol. 17, No. 6, 1984, pp. 607–619.
- [220] Lowe, D., *Perceptual organization and visual recognition*, Ph.D. thesis, Stanford Univ., San Francisco, CA, Sept 1984.
- [221] Pinjo, Z., Cyganski, D., and Orr, J., “Determination of 3D object orientation from projections,” *Pattern Recognition Letters*, Vol. 3, 1985, pp. 351–356.

- [222] Thompson, D. and Mundy, J., “Three dimensional model matching from an unconstrained viewpoint,” *IEEE Intl. Conf. on Robotics and Automation*, Piscataway, NJ, 1987.
- [223] Gavrila, D. and Groen, F., “3D object recognition from 2D images using geometric hashing,” *Pattern Recognition Letters*, Vol. 13, 1992, pp. 263–278.
- [224] Grimson, W. and Huttenlocher, D., “On the verification of hypothesized matches in model based recognition,” *IEEE Trans. on Pattern Analysis Machine Intelligence*, Vol. 13, No. 12, 1991, pp. 1201–1213.
- [225] Jurie, F., “Solution of the simultaneous pose and correspondence problem using Gaussian error model,” *Computer Vision and Image Understanding*, Vol. 73, No. 3, Mar 1999, pp. 357–373.
- [226] Sanchez-Gestido, M., “Computer vision methods for relative pose estimation and vision based navigation in rendezvous manoeuvres, on-orbit service operations and debris removal,” *Symp. on Advanced Space Technologies in Robotics and Automation*, Noordwijk, Netherlands, Nov 2015.
- [227] Post, M., Li, J., and Clark, C., “Visual pose estimation system for autonomous rendezvous of spacecraft,” *Symp. on Advanced Space Technologies in Robotics and Automation*, Noordwijk, The Netherlands, May 2015.
- [228] Zhang, G., Kontitsis, M., Filipe, N., Tsiotras, P., and Vela, P., “Cooperative relative navigation for space rendezvous and proximity operations using controlled active vision,” *Journal of Field Robotics*, Vol. 00, No. 0, 2015, pp. 1–24.
- [229] Harris, C. and Stephens, M., “A combined corner and edge detector,” *Proc. of the 4th Alvey Vision Conf.*, Vol. 15, 1988, pp. 147–151.
- [230] Rosten, E. and Drummond, T., “Machine learning for high-speed corner detection,” *European Conf. on Computer Vision*, 2006, pp. 430–443.
- [231] Diaz, J. and Abderrahim, M., “Visual inspection system for autonomous robotic on-orbit satellite servicing,” *Proc. of ESA Workshop on Advanced Space Technologies for Robotics and Automation*, Noordwijk, Netherlands, Nov 2006.
- [232] Arantes Jr., G., *Rendezvous with a non-cooperating target*, Master’s thesis, The Dept. of Production Engineering, Univ. of Bremen, Bremen, Germany, 2011.
- [233] Rublee, E., Rabaud, V., Konolige, K., and Bradski, G., “ORB: an efficient alternative to SIFT or SURF,” *IEEE Intl. Conf. on Computer Vision*, 2011.
- [234] Muja, M. and Lowe, D., “Scalable nearest neighbor algorithms for high dimensional data,” *IEEE Trans. on Pattern Analysis Machine Intelligence*, Vol. 36, No. 11, 2014, pp. 2227–2240.

- [235] Lepetit, V., “EPnP: an accurate $O(n)$ solution to the PnP problem,” *Intl. Journal of Computer Vision*, Vol. 81, No. 2, 2009, pp. 155–166.
- [236] Chen, J., Luo, X., Liu, H., and Sun, F., “Cognitively inspired 6D motion estimation of a noncooperative target using monocular RGB-D images,” *Cognitive Computation*, Vol. 8, 2016, pp. 1–9.
- [237] Sharma, S. and D’Amico, S., “Comparative assessment of techniques for initial pose estimation using monocular vision,” *Acta Astronautica*, Vol. 123, Jun–Jul 2016, pp. 435–445.
- [238] Cho, D., Tsiotras, P., Zhang, G., and Holzinger, M., “Robust feature detection, acquisition and tracking for relative navigation in space with a known target,” *Proc. of the AIAA Guidance, Navigation, and Control Conf.*, Boston, MA., Aug 2013.
- [239] Lourakis, M. and Zabulis, X., “Model-based visual tracking of orbiting satellites using edges,” *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems*, Vancouver, Canada, Sep 2017, pp. 3791–3796.
- [240] Cropp, A., Palmer, P., McLauchlan, P., and Underwood, C., “Estimating pose of known target satellite,” *Electronics Letters*, Vol. 36, No. 15, 2000, pp. 1331–1332.
- [241] Oumer, N. and Giorgio, P., “Tracking and pose estimation of non-cooperative satellite for OOS,” *Proc. of Intl. Symp. on Artificial Intelligence and Robotics and Automation in Space*, Turin, Italy, Sept 2012.
- [242] Lavagna, M., Pesce, V., and Bevilacqua, R., “Uncooperative objects pose, motion and inertia tensor estimation via stereo vision,” *Symp. on Advanced Space Technologies in Robotics and Automation*, Noordwijk, Netherlands, Nov 2015.
- [243] Sharma, S. and D’Amico, S., “Reduced-dynamics pose estimation for non-cooperative spacecraft rendezvous using monocular vision,” *AAS Guidance and Control Conf.*, Breckenridge, CO, Feb 2017, AAS 17-073.
- [244] Hartley, R. and Zisserman, A., *Multiple view geometry in computer vision*, Cambridge University Press, 2000.
- [245] Lu, C., Hager, G., and Mjolsness, E., “Fast and globally convergent pose estimation from video images,” *IEEE Trans. on Pattern Analysis Machine Intelligence*, Vol. 22, No. 6, 2000, pp. 610–622.
- [246] Schweighofer, G. and Pinz, A., “Globally optimal $O(n)$ solution to the PnP problem for general camera models,” *British Machine Vision Conf.*, Leeds, UK, Sept 2008.
- [247] Fan, B., Du, Y., and Cong, Y., “Robust and accurate online pose estimation algorithm via efficient 3D collinearity model,” *The Institution of Engineering and Technology Computer Vision*, Vol. 7, No. 5, 2013, pp. 382–393.

- [248] Zhou, H., Zhang, T., and Lu, W., “Vision-based pose estimation from points with unknown correspondences,” *IEEE Trans. on Image Processing*, Vol. 23, No. 8, 2014, pp. 3468–3478.
- [249] Sun, P., Sun, C., Li, W, Q., and P., W., “A new pose estimation algorithm using a perspective-ray-based scaled orthographic projection with iteration,” *PLOS ONE*, Vol. 10, No. 7, 2015, pp. e0134029.
- [250] Urban, S., Leitloff, J., and Hinz, S., “MLPnP-a real-time maximum likelihood solution to the perspective-n-point problem,” *arXiv:1607.08112*, 2016.
- [251] Gao, X., Hou, X., Tang, J., and Cheng, H., “Complete solution classification for the perspective-three-point problem,” *IEEE Trans. on Pattern Analysis Machine Intelligence*, Vol. 25, No. 8, 2003, pp. 930–943.
- [252] Kneip, L., Scaramuzza, D., and Siegwart, R., “A novel parameterization of the perspective-three-point problem for a direct computation of absolute camera position and orientation,” *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, Colorado Springs, CO, Jun 2011.
- [253] Moreno-Noguer, F., Lepetit, V., and Fua, P., “Accurate non-iterative $O(n)$ solution to the PnP problem,” *IEEE Intl. Conf. on Computer Vision*, IEEE, 2007, pp. 1–8.
- [254] Gao, J. and Zhang, Y., “An Improved Iterative Solution to the PnP Problem,” *Intl. Conf. on Virtual Reality and Visualization*, 2013.
- [255] Ferraz, L., Binefa, X., and Moreno-Noguer, F., “Very fast solution to the PnP problem with algebraic outlier rejection,” *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, 2014, pp. 501–508.
- [256] Li, S., Xu, C., and Xie, M., “A Robust $O(n)$ solution to the perspective-n-point problem,” *IEEE Trans. on Pattern Analysis Machine Intelligence*, Vol. 34, No. 7, 2012, pp. 1444–1450.
- [257] Zheng, Y., Sugimoto, S., and Okutomi, M., “ASPnP: An accurate and scalable solution to the perspective-n-point problem,” *Trans. on Information and Systems*, Vol. 96, No. 7, 2013, pp. 1525–1535.
- [258] Zheng, Y., Kuang, Y., Sugimoto, S., Aström, K., and Okutomi, M., “Revisiting the pnp problem: A fast, general and optimal solution,” *IEEE Intl. Conf. on Computer Vision*, IEEE, 2013, pp. 4321–4328.
- [259] Kneip, L. and Furgale, P., “UPnP: An optimal $O(n)$ solution to the absolute pose problem with universal applicability,” *European Conf. on Computer Vision*, IEEE, 2014, pp. 127–142.

- [260] Nakano, G., “Globally optimal DLS method for PnP problem with Cayley parameterization,” *British Machine Vision Conf.*, Swansea, UK, Sept 2015.
- [261] Hesch, J. and Roumeliotis, S., “A Direct Least-Squares (DLS) method for PnP,” *IEEE Intl. Conf. on Computer Vision*, Barcelona, Spain, Nov 2011.
- [262] Dementhon, D. and Davis, L., “Model-based object pose in 25 lines of code,” *Intl. Journal of Computer Vision*, Vol. 15, No. 1, 1995, pp. 123–141.
- [263] Gold, S., Rangarajan, A., Lu, C. P., Pappu, S., and Mjolsness, E., “New algorithms for 2D and 3D point matching: pose estimation and correspondence,” *Pattern Recognition*, Vol. 31, No. 8, 1998, pp. 1019–1031.
- [264] Kirkpatrick, S., Gelatt, C., and Vecchi, M., “Optimization by simulated annealing,” *Science, New Series*, Vol. 220, No. 4598, 1983, pp. 671–680.
- [265] Choi, C. and Christensen, H., “RGB-D object pose estimation in unstructured environments,” *Robotics and Autonomous System*, Vol. 75, 2016, pp. 595–613.
- [266] Rosenhahn, B., Brox, T., and Weickert, J., “Three-dimensional shape knowledge for joint image segmentation and pose tracking,” *Intl. Journal of Computer Vision*, Vol. 73, No. 3, 2007, pp. 243–262.
- [267] Kass, M., Witkin, A., and Terzopoulos, D., “Snakes: active contour models,” *Intl. Journal of Computer Vision*, Vol. 1, No. 4, 1988, pp. 321–441.
- [268] Mumford, D. and Shah, J., “Optimal approximations by piecewise smooth functions and associated variational problems,” *Commun. Pure Appl. Math*, Vol. 42, 1989, pp. 577–685.
- [269] Caselles, V., Kimmel, R., and Sapiro, G., “Geodesic active contours,” *Intl. Journal of Computer Vision*, Vol. 22, No. 1, 1997, pp. 61–79.
- [270] Chan, T. and Vese, L., “Active contours without edges,” *IEEE Tran. on Image Processing*, 2001, pp. 266–277.
- [271] Osher, S. and Sethian, J., “Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations,” *Journal of Computational Physics*, Vol. 79, No. 1, 1988, pp. 12–49.
- [272] Kervrann, C. and Heitz, F., “Statistical deformable model-based segmentation of image motion,” *IEEE Trans. on Image Processing*, Vol. 8, No. 4, 1999, pp. 583–588.
- [273] Zhu, S., T.S., L., and Yuille, A., “Region competition: unifying snakes, region growing, energy /Bayes/MDL for multi-band image segmentation,” *IEEE Trans. on Pattern Analysis Machine Intelligence*, Vol. 18, No. 9, 1996, pp. 884–900.

- [274] Leventon, M., Grimson, W., and Faugeras, O., “Statistical shape influence in geodesic active contours,” *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, Vol. 1, Hilton Head Island, SC, 2000, pp. 316–323.
- [275] Rousson, M. and Paragios, N., “Shape priors for level set representations,” *European Conf. on Computer Vision*, 2002, pp. 78–92.
- [276] Cremers, D., *Statistical Shape Knowledge in Variational Image Segmentation*, Ph.D. thesis, Dept. of Mathematics and Computer Science, University of Mannheim, Mannheim, Germany, 2002.
- [277] Cremers, D. and Schnörr, C., “Motion competition: variational integration of motion segmentation and shape regularization,” *Pattern Recognition*, Vol. 2449, 2002, pp. 472–480.
- [278] Brox, T. and Weickert, J., “Level set based image segmentation with multiple regions,” *Lecture Notes in Computer Science*, Aug 2004, pp. 415–423.
- [279] Bibby, C. and I., R., “Robust real-time visual tracking using pixel-wise posteriors,” *European Conf. on Computer Vision*, Berlin, Heidelberg, 2008, pp. 831–844.
- [280] Rosenhahn, B., T., B., D., C., and Seidel, H., “A comparison of shape matching methods for contour based pose estimation,” *Combinatorial Image Analysis*, 2006, pp. 263–276.
- [281] Brox, T., Rosenhahn, B., Gall, J., and Cremers, D., “Combined region and motion-based 3D tracking of rigid and articulated objects,” *IEEE Trans. on Pattern Analysis Machine Intelligence*, Vol. 32, No. 3, 2010, pp. 402–415.
- [282] Schmaltz, C. and Rosenhahn, B., “Region-based pose tracking with occlusions using 3D models,” *Machine Vision and Applications*, Vol. 23, No. 3, 2012, pp. 557–577.
- [283] Hexner, J. and Hagege, R., “2D-3D pose estimation of heterogeneous objects using a region based approach,” *Intl. Journal of Computer Vision*, Vol. 118, No. 1, 2016, pp. 95–112.
- [284] Tjaden, H., Schwanecke, U., and Schömer, E., “Real-time monocular pose estimation of 3D objects using temporally consistent local color histograms,” *IEEE Intl. Conf. on Computer Vision*, 2017, pp. 124–132.
- [285] Li, C., Xu, C., Gui, C., and Fox, M., “Distance regularized level set evolution and its application to image segmentation,” *IEEE Trans. on Image Processing*, Vol. 19, No. 12, 2010, pp. 3243–3254.

- [286] Dambreville, S., Sandhu, R., Yezzi, A., and Tannenbaum, A., “A geometric approach to joint 2D region-based segmentation and 3D pose estimation using a 3D shape prior,” *Journal on Imaging Sciences, Society for Industrial and Applied Mathematics*, Vol. 3, No. 1, 2010, pp. 110–132.
- [287] Jayawardena, S., Ying, D., and Hutter, M., “3D model assisted image segmentation,” *Intl. Conf. on Digital Image Computing Techniques and Applications*, 2011, pp. 51–58.
- [288] Prisacariu, V. and Reid, I., “PWP3D: real-time segmentation and tracking of 3D objects,” *British Machine Vision Conf.*, Sept 2009.
- [289] Prisacariu, V. and Reid, I., “Shared shape spaces,” *IEEE Intl. Conf. on Computer Vision*, 2011, pp. 2587–2594.
- [290] Prisacariu, V. and Reid, I., “Nonlinear shape manifolds as shape priors in level set segmentation and tracking,” *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, 2011, pp. 2185–2192.
- [291] Gong, J., Fan, G., Yu, L., J., H., Chen, D., and Fan, N., “Joint target tracking, recognition and segmentation for infrared imagery using a shape manifold-based level set,” *Sensors*, Vol. 14, 2014, pp. 10124–10145.
- [292] Zhao, S., Wang, L., Sui, W., Wu, H., and Pan, C., “3D object tracking via boundary constrained region-based model,” *IEEE Intl. Conf. on Image Processing*, Paris, France, Oct 2014, pp. 486–490.
- [293] Perez-Yus, A., Puig, L., Lopez-Nicolas, G., Guerrero, J., and Fox, D., “RGB-D based tracking of complex objects,” *Intl. Workshop on Understanding Human Activities Through 3D Sensors*, Cancún, México, Dec 2015.
- [294] Prisacariu, V., Kähler, O., Murray, D., and Reid, I., “Real-time 3D tracking and reconstruction on mobile phones,” *IEEE Trans. on Visualization and Computer Graphics*, Vol. 21, No. 5, 2015, pp. 557–570.
- [295] Tjaden, H., Schwanecke, U., and Schömer, E., “Real-time monocular segmentation and pose tracking of multiple objects,” *European Conf. on Computer Vision*, 2016, pp. 423–438.
- [296] Swierczynski, P., Papiez, B., Schnabel, J., and Macdonald, C., “A level-set approach to joint image segmentation and registration with application to CT lung imaging,” *Computerized Medical Imaging and Graphics*, Jun 2017.
- [297] Shi, J., Ulrich, S., and Ruel, S., “Real-time saliency detection for grayscale and colour images,” *Intl. Journal of Computer Vision*.

- [298] Yan, Q., Xu, L., Shi, J., and Jia, J., “Hierarchical saliency detection,” *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, 2013.
- [299] Shi, J., Yan, Q., Xu, L., and Jia, J., “Hierarchical image saliency detection on extended CSSD,” *IEEE Trans. on Pattern Analysis Machine Intelligence*, Vol. 38, No. 4, 2016, pp. 717–729.
- [300] Shi, J., Ulrich, S., and Ruel, S., “Regional method for monocular infrared image spacecraft pose estimation,” *Proc. of the AIAA Space Conf. and Exhibit*, Orlando, FL, Sept 2018.
- [301] Shi, J., Ulrich, S., and Ruel, S., “Saliency detection and 6-DOF pose estimation of monochromatic monocular spacecraft images,” *IEEE Trans. on Image Processing*.
- [302] Shi, J., Ulrich, S., and Ruel, S., “An unsupervised method of infrared spacecraft image foreground extraction.” *AIAA Journal of Spacecraft and Rockets*.
- [303] Shi, J., Ulrich, S., and Ruel, S., “International space station image extraction from a dynamic environment using deep learning,” *Proc. of the Intl. Conf. on Control, Dynamic Systems, and Robotics*, Toronto, Canada, Aug 2017.
- [304] Shi, J., Ulrich, S., and Ruel, S., “Level-set and image statistics for pose estimation of satellites,” *Proc. of the Intl. Conf. on Control, Dynamics Systems, and Robotics*, Niagara Falls, Canada, June 2018.
- [305] Shi, J., Ulrich, S., and Ruel, S., “A comparison of feature descriptors using monocular thermal camera images,” *Proc. of the Intl. Conf. on Control, Automation and Robotics*, Nagoya, Japan, Apr 2017.
- [306] Shi, J., Ulrich, S., and Ruel, S., “yBRIEF: a study of non-Gaussian binary elementary features,” *Proc. of the Intl. Conf. on Image, Vision and Computing*, Chengdu, China, June 2017.
- [307] Shi, J., Ulrich, S., and Ruel, S., “CubeSat simulation and detection using monocular camera images and convolutional neural networks,” *Proc. of the AIAA Guidance, Navigation, and Controls Conf. and Exhibit*, Kissimmee, FL, Jan 2018.
- [308] Shi, J., Ulrich, S., and Ruel, S., “Uncooperative spacecraft pose estimation using an infrared camera during proximity operations,” *Proc. of the AIAA Space Conf. and Exhibit*, Pasadena, CA., Aug-Sept 2015.
- [309] Shi, J., Ulrich, S., and Ruel, S., “Spacecraft pose estimation using a monocular camera,” *Proc. of the Intl. Astronautical Congress*, Guadalajara, México, Sept 2016.
- [310] Shi, J. and Ulrich, S., “SoftPOSIT enhancements for monocular camera spacecraft pose estimation,” *IEEE Intl. Conf. on Methods and Models in Automation and Robotics*, Międzyzdroje, Poland, Aug 2016, DOI: 10.1109/MMAR.2016.7575083.

- [311] Tomasi, C. and Manduchi, R., “Bilateral filtering for gray and color images,” *Intl. Conf. on Computer Vision*, Bombay, India, 1998.
- [312] Srinivasan, S. and Balram, N., “Adaptive contrast enhancement using local region stretching,” *Proc. of Asian Symp. on Information Display*, New Delhi, India, 2006, pp. 152–155.
- [313] Kwok, N., Wang, D., Ha, P., Fang, G., and Chen, S., *Locally-equalized image contrast enhancement using PSO-tuned sectorized equalization*, Springer-Verlag Berlin Heidelberg, 2012.
- [314] Lucas, J., Calef, B., and Knox, K., “Image enhancement for astronomical scenes,” *SPIE Optical Engineering and Applications*, Sep 2013, p. 885603.
- [315] Pizer, S., Auston, J., Perry, J., and Safrit, H., “Adaptive histogram equalization for automatic contrast enhancement of medical images,” *Proc. of SPIE XIV/PACS IV Conf. Medicine*, Newport Beach, CA, 1986.
- [316] Pizer, S., Amburn, E., Austin, J., Cromartie, R., Geselowitz, A., Greer, T., Romeny, B., Zimmerman, J., and Zuiderveld, K., “Adaptive histogram equalization and its variations,” *Computer Vision, Graphics, and Image Processing*, Vol. 39, 1987, pp. 355–368.
- [317] Zuiderveld, K., “Contrast limited adaptive histogram equalization,” *Graphics Gems IV*, 1994, pp. 474–485.
- [318] Kim, S., Min, B., Lim, D., and Lee, J., “Determining parameters in contrast limited adaptive histogram equalization,” *Proc. of the Intl. Conf. on Information Security and Assurance*, Vol. 21, 2013, pp. 204–207.
- [319] Min, B., Lim, D., Kim, S., and Lee, J., “A novel method of determining parameters of CLAHE based on image entropy,” *Intl. Journal of Software Engineering and Its Applications*, Vol. 7, No. 5, 2013, pp. 113–120.
- [320] Muniyappan, S., Allirani, A., and Saraswathi, S., “A novel approach for image enhancement by using contrast limited adaptive histogram equalization method,” *Intl. Conf. on Computing, Communications and Networking Technologies*, Tiruchengode, India, 2013.
- [321] Sobel, I. and Feldman, G., “A 3x3 isotropic gradient operator for image processing,” *Stanford Artificial Intelligence Project*, 1968.
- [322] Scharf, H., *Optimal operators in digital image processing*, Ph.D. thesis, Rupertus Carola University of Heidelberg, Heidelberg, Germany, 2000.
- [323] Roberts, L., *Machine perception of 3D solids*, Optical and Electro-optical Information Processing, MIT Press, 1965.

- [324] Canny, J., “A computational approach to edge detection,” *IEEE Trans. on Pattern Analysis Machine Intelligence*, Vol. 8, No. 6, 1986, pp. 679–698.
- [325] Shi, J. and Tomasi, C., “Good features to track,” *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, Seattle, WA, Jun 1994.
- [326] Matas, J., Chum, O., and Urban, M. Pajdla, T., “Robust wide baseline stereo from maximally stable extremal regions,” *British Machine Vision Conf.*, 2002.
- [327] Mair, E., Hager, G., Burschka, D., Suppa, M., and Hirzinger, G., “Adaptive and generic corner detection based on the accelerated segment test,” *European Conf. on Computer Vision*, Heraklion, Crete, Sept 2010.
- [328] Agrawal, M., Konolige, K., and Blas, M., “CenSurE: center surround extremas for realtime feature detection and matching,” *European Conf. on Computer Vision*, Berlin, Heidelberg, Oct 2008, pp. 102–115.
- [329] Tombari, F. and Stefano, L., “Interest points via maximal self-dissimilarities,” *Asian Conf. on Computer Vision*, Springer International Publishing, Singapore, Nov 2014.
- [330] Tola, E., Lepetit, V., and Fua, P., “DAISY: an efficient dense descriptor applied to wide-baseline stereo,” *IEEE Trans. on Pattern Analysis Machine Intelligence*, Vol. 32, No. 5, 2010, pp. 815–830.
- [331] Ziegler, A., Christiansen, E., Kriegman, D., and Belongie, S., “Locally uniform comparison image descriptor,” *Neural Information Processing Systems*, 2012.
- [332] Levi, G. and Hassner, T., “LATCH: learned arrangements of three patch codes,” *IEEE Winter Conf. on Applications of Computer Vision*, 2016.
- [333] Lowe, D., “Object recognition from local scale-invariant features,” *IEEE Intl. Conf. on Computer Vision*, Kerkyra, Greece, Sept 1999.
- [334] Wilson, H. and Bergen, J., “A four mechanism model for spatial vision,” *Vision Research*, Vol. 19, 1979, pp. 19–32.
- [335] Burt, P., “Fast filter transforms for image processing,” *Computer Vision, Graphics and Image Processing*, Vol. 16, 1981, pp. 20–51.
- [336] Mikolajczyk, K., *Detection of local features invariant to affine transformations*, Ph.D. thesis, Institut National Polytechnique de Grenoble, Grenoble, France, 2002.
- [337] Noble, A., *Descriptions of image surfaces*, Ph.D. thesis, Oxford Univ., Oxford, UK, 1989.
- [338] Smith, S. and Brady, J., “SUSAN-a new approach to low level image processing,” *Intl. Journal of Computer Vision*, Vol. 23, 1997, pp. 45–78.

- [339] Rosten, E. and Drummond, T., “Fusing points and lines for high performance tracking,” *IEEE Intl. Conf. on Computer Vision*, Beijing, China, Oct 2005.
- [340] Mikolajczyk, K. and Schmid, C., “Indexing based on scale invariant interest points,” *IEEE Intl. Conf. on Computer Vision*, 2001.
- [341] Rosin, P., “Measuring corner properties,” *Computer Vision and Image Understanding*, Vol. 73, No. 2, 1999, pp. 291–307.
- [342] Alcantarilla, P., Bartoli, A., and Davison, A., “KAZE features,” *European Conf. on Computer Vision*, 2012.
- [343] Perona, P. and Malik, J., “Scale-space and edge detection using anisotropic diffusion,” *IEEE Trans. on Pattern Analysis Machine Intelligence*, Vol. 12, No. 7, 1990, pp. 629–639.
- [344] Yang, X. and Cheng, K., “LDB: An ultra-fast feature for scalable augmented reality,” *IEEE and ACM Intl. Symp. on Mixed and Augmented Reality*, 2012.
- [345] Schmidt, A., Kraft, M., and Kasinowski, A., *An evaluation of image feature detectors and descriptors for robot navigation*, Computer Vision and Graphics, Springer Berlin Heidelberg, 2010.
- [346] Kimmel, R., Zhang, C., Bronstein, A., and Bronstein, M., “Are MSER features really interesting?” *IEEE Trans. on Pattern Analysis Machine Intelligence*, Vol. 33, No. 11, 2011, pp. 2316–2320.
- [347] Ojala, T., Pietikäinen, M., and Mäenpää, T., “A generalized local binary pattern operator for multiresolution gray scale and rotation invariant texture classification,” *Intl. Conf. on Advances in Pattern Recognition*, 2001.
- [348] Wolf, L., Hassner, T., and Taigman, Y., “Effective unconstrained face recognition by combining multiple descriptors and learned background statistics,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 33, No. 10, 2011, pp. 1978–1990.
- [349] Mikolajczyk, K. and Schmid, C., “A performance evaluation of local descriptors,” *IEEE Trans. on Pattern Analysis Machine Intelligence*, Vol. 27, No. 10, 2005, pp. 1615–1630.
- [350] Muja, M. and Lowe, D., “Fast approximate nearest neighbors with automatic algorithm configuration,” *Intl. Conf. on Computer Vision Theory and Applications*, Vol. 1, 2009.
- [351] Fawcett, T., “An introduction to ROC analysis,” *Pattern Recognition Letters*, Vol. 27, 2006, pp. 861–874.
- [352] Mikolajczyk, K. and Schmid, C., “An affine invariant interest point detector,” *European Conf. on Computer Vision*, Copenhagen, Denmark, May 2002, pp. 128–142.

- [353] Kadir, T. and Brady, M., “Saliency, scale and image description,” *Intl. Journal of Computer Vision*, Vol. 45, No. 2, 2001, pp. 83–105.
- [354] Agarwal, S. and Roth, D., “Learning a sparse representation for object detection,” *European Conf. on Computer Vision*, Copenhagen, Denmark, May 2002, pp. 113–127.
- [355] Leibe, B., Leonardis, A., and Schiele, B., “Combined object categorization and segmentation with an implicit shape model,” *European Conf. on Computer Vision*, Vol. 2, Prague, Czech Republic, May 2004, pp. 7–23.
- [356] Duda, R., *Pattern classification*, John Wiley and Son, 2nd ed., 2012.
- [357] Bentley, J., “Multidimensional binary search trees used for associative searching,” *Communications of the ACM*, Vol. 18, No. 9, 1975, pp. 509–517.
- [358] Andoni, A. and Indyk, P., “Near-optimal hashing algorithms for approximate nearest neighbour in high dimensions,” *Communications of the ACM*, Vol. 51, No. 1, 2008, pp. 117–122.
- [359] Babenko, A. and Lempitsky, V., “The inverted multi-index,” *IEEE Trans. on Pattern Analysis Machine Intelligence*, Vol. 37, No. 6, 2015, pp. 1247–1260.
- [360] Babenko, A. and Lempitsky, V., “Efficient indexing of billion-scale datasets of deep descriptors,” *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, 2016.
- [361] Manning, C., Raghavan, P., and Schütze, H., *Introduction to information retrieval*, Cambridge University Press, 1st ed., 2008.
- [362] Everingham, M., L., V. G., Williams, C., Winn, J., and Zisserman, A., “The pascal Visual Object Classes (VOC) challenge,” *Intl. Journal of Computer Vision*, Vol. 88, No. 2, 2010, pp. 303–338.
- [363] Lin, T., Marie, M., Belongie, J., Hays, P., Perona, D., Ramanan, D., Dollár, P., and Zitnick, L., “Microsoft COCO: common objects in context,” *European Conf. on Computer Vision*, 2014.
- [364] Sermanet, P., Eigen, D., Zhang, X., Mathieu, M. Fergus, R., and LeCun, Y., “OverFeat: integrated recognition, localization and detection using convolutional networks,” *arXiv preprint*, arXiv:1312.6229, 2013.
- [365] Ioffe, S. and Szegedy, C., “Batch normalization: accelerating deep network training by reducing internal covariate shift,” *arXiv preprint*, arXiv:1502.03167, 2015.
- [366] Maas, A., Hannun, A., and Ng, A., “Rectifier nonlinearityies improve neural network acoustic models,” *Intl. Conf. on Machine Learning*, Vol. 30, No. 1, 2013.

- [367] Nair, V. and Hinton, G., “Rectified linear units improve restricted boltzmann machines,” *Intl. Conf. on Machine Learning*, 2010.
- [368] Hinton, G., Srivastava, N., Krizhevsky, A., and Sutskever, I. Salakhutdinov, R., “Improving neural networks by preventing co-adaptation of feature detectors,” *arXiv preprint*, arXiv:1207.0580, 2012.
- [369] Krizhevsky, A. and Hinton, G., “Learning multiple layers of features from tiny images,” *Univ. of Toronto Tech. Report*, 2009.
- [370] Lin, M., Chen, Q., and Yan, S., “Network in network,” *Computing Research Repository*, abs/1312.4400, 2013.
- [371] Radford, A., Metz, L., and Chintala, S., “Unsupervised representation learning with deep convolutional generative adversarial networks,” *arXiv preprint*, arXiv:1511.06434, 2015.
- [372] Fei-fei, L., Fergus, R., and Perona, P., “One-shot learning of object categories,” *IEEE Trans. on Pattern Analysis Machine Intelligence*, Vol. 28, No. 4, 2006, pp. 594–611.
- [373] Zhang, L., Yang, C., Lu, H., Ruan, X., and Yang, M., “Ranking saliency,” *IEEE Trans. on Pattern Analysis Machine Intelligence*, Vol. 39, No. 9, 2017, pp. 1892–1904.
- [374] Felzenszwalb, P. and Huttenlocher, D., “Efficient graph-based image segmentation,” *Intl. Journal of Computer Vision*, Vol. 59, No. 2, 2004, pp. 167–181.
- [375] Rother, C., Kolmogorov, V., and Blake, A., “GrabCut-interactive foreground extraction using iterated graph cuts,” *ACM Trans. on Graphics*, Vol. 23, No. 3, 2004, pp. 309–314.
- [376] Strand, R., Ciesielski, K., Malmberg, F., and Saha, P., “The minimum barrier distance,” *Computer Vision and Image Understanding*, Vol. 117, No. 4, 2013, pp. 429–437.
- [377] Zhou, D., Weston, J., and Gretton, A., “Ranking on data manifolds,” *Neural Information Processing Systems*, 2004.
- [378] Wan, X., Yang, J., and Xiao, J., “Manifold-ranking based topic-focused multi-document summarization,” *Proc. of Intl. Joint Conf. on Artificial Intelligence*, 2007, pp. 2903–2908.
- [379] Xu, B., Bu, J., Chen, C., Cai, D., He, X., Liu, W., and Luo, J., “Efficient manifold ranking for image retrieval,” *ACM Special Interest Group on Information Retrieval*, 2011.
- [380] Fujiwara, Y., Irie, G., Kuroyama, S., and Onizuka, M., “Scaling manifold ranking based image retrieval,” *Proc. of the VLDB Endowment*, Vol. 8, 2014.

- [381] Wang, J., Lu, H., Li, X., Tong, N., and Liu, W., “Saliency detection via background and foreground seed selection,” *Neurocomputing*, Vol. 152, 2015, pp. 359–368.
- [382] Cheng, M., Zhang, G., Mitra, N., Huang, X., and Hu, S., “Global contrast based salient region detection,” *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, 2011, pp. 409–416.
- [383] Meyer, F., “Color image segmentation,” *IET Intl. Conf. on Image Processing and its Applications*, 1992.
- [384] Cheng, M., Zhang, Z., Lin, W., and Torr, P., “BING: binarized normed gradients for objectness estimates at 300 fps,” *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, 2014, pp. 3286–3293.
- [385] Hubel, D. and Wiesel, T., “Receptive fields, binocular interaction and functional architecture in the cat’s visual cortex,” *Journal of Physiology*, Vol. 160, No. 1, 1962, pp. 106–154.
- [386] Honeine, P. and Richard, C., “Preimage problem in kernel-based machine learning,” *IEEE Signal Processing Magazine*, Vol. 28, No. 2, 2011, pp. 77–88.
- [387] Jolliffe, I., *Principal component analysis*, Springer, New York, 2nd ed., 2002.
- [388] Pearson, K., “On lines and planes of closest fit to systems of points in space,” *Phil. Mag*, Vol. 6, No. 2, 1901, pp. 559–572.
- [389] Hotelling, H., “Analysis of a complex of statistical variables in to principal componenets,” *Journal of Edu. Psychol.*, Vol. 24, 1933, pp. 417–441,498–520.
- [390] Sirovich, L. and Kirby, M., “A low dimensional procedure for the characterization of human faces,” *Journal of the Optical Society of America A*, Vol. 4, No. 3, 1987, pp. 519–524.
- [391] Turk, M. and Pentland, A., “Eigenfaces for recognition,” *Journal of Cognitive Neuroscience*, Vol. 3, No. 1, 1991, pp. 71–86.
- [392] Javed, A., “Face recognition based on principal component analysis,” *Intl. Journal of Image, Graphics and Signal Processing*, Vol. 5, No. 2, 2013, pp. 38–44.
- [393] Cavalcanti, G., Ren, T., and Pereira, J., “Weighted modular image principal component analysis for face recognition,” *Expert Systems with Applications*, Vol. 40, No. 12, 2013, pp. 4971–4977.
- [394] Mena-Chalco, J., Veihö, L., Macêdo, I., and Cesar-Jr., R., “PCA-based 3D face photography,” *IEEE XXI Brazilian Symp. on Computer Graphics and image Processing*, Oct 2008, pp. 313–320.

- [395] Mena-Chalco, J., Macêdo, I., and Veiho, L., “3D face computational photography using PCA spaces,” *The Visual Computer*, Vol. 25, No. 10, 2009, pp. 899–909.
- [396] Du, H., Hu, Q., Jiang, M., and Zhang, F., “Two-dimensional principal component analysis based on Schatten p-norm for image feature extraction,” *Journal of Visual Communication and Image Representation*, Vol. 32, 2015, pp. 55–62.
- [397] Wang, J., “Generalized 2D principal component analysis by Lp-Norm for image analysis,” *IEEE Trans. on Cybernetics*, Vol. 46, No. 3, 2016, pp. 792–803.
- [398] Schölkopf, B., Smola, A., and K.R., M., “Nonlinear component analysis as a kernel,” *Neural Computation*, Vol. 10, No. 5, 1998, pp. 1299–1319.
- [399] Mika, S., Schölkopf, B., Smola, A., Müller, K., Scholz, M., and Rätsch, G., “Kernel PCA and de-noising in feature spaces,” *Neural Information Processing Systems*, Vol. 11, No. 1, 1999, pp. 536–542.
- [400] la Torre, F. and Nguyen, M., “Parameterized kernel principal component analysis: theory and applications to supervised and unsupervised image alignment,” *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, IEEE, 2008.
- [401] Simon, D., *Fast and accurate shape-based registration*, Ph.D. thesis, Carnegie Mellon University, Pittsburgh, Pennsylvania, 1996.
- [402] Mark, L., *Principal component analysis for ICP pose estimation of space structures*, Master’s thesis, The Dept. of Aerospace Engineering, Ryerson Univ., Toronto, Canada, 2010.
- [403] Ke, Y. and Sukthankar, R., “PCA-SIFT: A more distinctive representation for local image descriptors,” *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, 2004, pp. 511–517.
- [404] Trucco, E. and Verri, A., *Introductory techniques for 3D computer vision*, Prentice-Hall, Inc, 1998.
- [405] Eckart, C. and Young, G., “The approximation of one matrix by another of lower rank,” *Psychometrika*, Vol. 1, No. 3, 1936, pp. 211–218.
- [406] Baglama, J. and Reichel, L., “Augmented implicitly restarted Lanczos bidiagonalization methods,” *Journal on Scientific Computing, Society for Industrial and Applied Mathematics*, Vol. 27, No. 1, 2005, pp. 19–42.
- [407] Larsen, R., “Lanczos bidiagonalization with partial reorthogonalization,” *DAIMI Report Series*, Vol. 27, No. 537, 1998.
- [408] Rokhlin, V., Szlam, A., and Tygert, M., “A randomized algorithm for principal component analysis,” *Journal on Matrix Analysis and Applications, Society for Industrial and Applied Mathematics*, Vol. 31, No. 3, 2009, pp. 1100–1124.

- [409] Halko, N., Martinsson, P., and J.A., T., "Finding structure with randomness: probabilistic algorithms for constructing approximate matrix decompositions," *Review Society for Industrial and Applied Mathematics*, Vol. 53, No. 2, 2009, pp. 217–288.
- [410] Boser, B., Guyon, I., and Vapnik, V., "A training algorithm for optimal margin classifiers," *Proc. of Conf. on Learning Theory*, ACM Press, Pittsburgh, PA, 1992.
- [411] Schölkopf, *Support vector learning*, Ph.D. thesis, Dept. Computer Science, Univ. of Berlin, Oldenbourg Verlag, Munich, 1997.
- [412] Fitch, A., Kadyrov, A., Christmas, W., and Kittler, J., "Fast robust correlation," *IEEE Trans. on Image Processing*, Vol. 14, No. 8, 2005, pp. 1063–1073.
- [413] De La Torre, F. and Black, M., "A framework for robust subspace learning," *Intl. Journal of Computer Vision*, Vol. 54, No. 1–3, 2004, pp. 117–142.
- [414] Kwak, N., "Principal component analysis based on L1-norm maximization," *IEEE Trans. on Pattern Analysis Machine Intelligence*, Vol. 30, No. 9, 2008, pp. 1672–1680.
- [415] He, R., Hu, B., Zheng, W., and Kong, X., "Robust principal component analysis based on maximum correntropy criterion," *IEEE Trans. on Image Processing*, Vol. 20, No. 6, 2011, pp. 1485–1494.
- [416] Ke, Q. and Kanade, T., "Robust L1 norm factorization in the presence of outliers and missing data by alternative convex programming," *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, IEEE, 2005.
- [417] Möller, T. and Trumbore, B., "Fast, minimum storage ray-triangle intersection," *Journal of Graphic Tools*, Vol. 2, No. 2, 1997, pp. 21–28.
- [418] Jimenez, J., Segura, R., and Feito, F., "A robust segment/triangle intersection algorithm for interference tests. Efficiency study," *Computational Geometry*, Vol. 43, No. 5, 2010, pp. 474–492.
- [419] Isack, H. and Boykov, Y., "Joint optimization of fitting and matching in multi-view reconstruction," *arXiv preprint*, 2013.
- [420] Sinkhorn, R., "A relationship between arbitrary positive matrices and doubly stochastic matrices," *The Annals of Mathematical Statistics*, Vol. 35, No. 2, 1964, pp. 876–879.
- [421] Jager, K., Hebel, M., and Bers, K., "Automatic 3D object pose estimation in IR image sequences for forward motion application," *Proc. of SPIE Automatic Target Recognition XIV*, Vol. 5426, 2004.
- [422] Diaz, J. and Abderrahim, M., "Modified SoftPOSIT algorithm for 3D visual tracking," *IEEE International Symp. on Intelligent Signal Processing*, oct 2007.

- [423] Ibe, O., *Markov process for stochastic modeling*, Elsevier Academic Press, 2007.
- [424] Challis, “A procedure for determining rigid body transformation parameters,” *Journal of Biomechanics*, Vol. 28, No. 6, 1995, pp. 733–737.
- [425] Tsai, A., A., Y., Wells, W., Tempany, C., D., T., Fan, A., Grimson, E., and Willsky, A., “Model-based curve evolution technique for image segmentation,” *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, Kauai, Hawaii, 2001, pp. 463–468.
- [426] Cremers, D., Rousson, M., and Deriche, R., “A review of statistical approaches to level set segmentation: integrating color, texture, motion and shape,” *Intl. Journal of Computer Vision*, Vol. 72, No. 2, 2007, pp. 195–215.
- [427] Nelder, J. and Mead, R., “A simplex method for function minimization,” *The Computer Journal*, Vol. 7, No. 4, 1965, pp. 308–313.
- [428] Hager, W. and Zhang, H., “A new conjugate gradient method with guaranteed descent and an efficient line search,” *SIAM Journal on Optimization*, Vol. 16, No. 1, 2005, pp. 170–192.
- [429] SSP-30219, “Space station reference coordinate systems,” Jun 2005, NASA International Space Station Program, Rev H.
- [430] Goodman, J., “Rendezvous and proximity operations of the space shuttle,” Jul 2005, NASA Technical Reports Server.
- [431] English, C., Okouneva, G., Saint-Cyr, P., Choudhuri, A., and Luu, T., “Real-time dynamic pose estimation systems in space: lessons learned for system design and performance evaluation,” *Intl. Journal of Intelligent Control and Systems*, Vol. 16, No. 2, 2011, pp. 79–96.
- [432] English, C., Okouneva, G., and Choudhuri, A., “Shape-based pose estimation evaluation using expectivity index artifacts,” *Workshop on Performance Metrics for Intelligent Systems*, 2012.
- [433] Gerald, C. and Wheatley, P., *Applied numerical analysis*, Addison-Wesley Publishing Company, 5th ed., 1994.
- [434] Shi, J. and Ulrich, S., “Spacecraft adaptive attitude control with application to space station free-flyer robotic capture,” *Proc. of the AIAA Guidance, Navigation, and Control Conf.*, Kissimmee, FL., Jan 2015.
- [435] Hughes, P., *Spacecraft attitude dynamics*, Dover Publication Inc., 2nd ed., 2004.

Appendix A: List of Videos

Analysis videos are available online. Experimental datasets and video images are available to download from <http://ai-automata.ca/research/hisafe.html>.

1	STS-135 ISS undocking pose estimation	https://youtu.be/U4xldh-YWos
2	STS-135 ISS docking infrared camera pose estimation	https://youtu.be/gJPGJF4vXoo
3	Enhanced gradient descent	https://youtu.be/Gm3LhfQ2_KM
4	Level-set evolution	https://youtu.be/TvITD6d386g
5	Envisat pose estimation	https://youtu.be/_EfyimKAKVY
6	STS-135 ISS pose estimation gradient descent	https://youtu.be/8s91TkZ__XM
7	RSM thermal image pose estimation	https://youtu.be/Sg4y_bAERgE
8	Gradient descent initialisation using FC-HSF	https://youtu.be/fqtMsGsnexE
9	STS-135 ISS image saliency foreground prediction	https://youtu.be/Jr_yEOzdNcg
10	Single frame pose convergence	https://youtu.be/9JVeCPvzuFo
11	ePnP vs. PWP3D pose estimation	https://youtu.be/8Km--FOmC8E
12	Lores vs hires internal model	https://youtu.be/IEMpdNHJwic
13	ISS lores/hires single frame convergence	https://youtu.be/gkKL_8PEcSI
14	ISS pose estimation	https://youtu.be/SsUuOYbmYlU
15	Inception-Resnet Faster-RCNN demo	https://youtu.be/GwaIWk6cjzs
16	Resnet-101 Faster-RCNN demo	https://youtu.be/MlceJWM4pKM
17	Cubesat detection and localisation using Faster-RCNN	https://youtu.be/AfBw4jGBz6Y
18	Background subtraction	https://youtu.be/B9ehT1Di_2w
19	Feature comparison	https://youtu.be/aHC-OZINI9k
20	Envisat pose estimation	https://youtu.be/wQN7BU8P0Y4
21	CubeSat ProxOps Visualisation and Camera Emulation	https://youtu.be/AozgJbjf4Ac

Appendix B: Dynamic Motion

A full orbit and attitude dynamic simulation environment is developed to generate realistic target and chaser spacecraft motion. We designate the target spacecraft as the Client Satellite (CS) and the chaser vehicle as the Servicer Spacecraft (SS). The input classical orbital elements are converted into the J2000 Earth Centric Inertial (J2000-ECI) frame denoted by \mathcal{F}_{00} , and the CS Local Vertical Local Horizontal (LVLH) frame is \mathcal{F}_{co} . We use the two body equation for orbit simulation of the CS and SS. We verify the relative motion against the Clohessy-Wiltshire (CW) solution. We convert the initial attitude state into quaternions and Modified Rodriguez Parameters (MRP). The attitude state propagation will be performed using MRP and the Runge-Kutta-4th order (RK4) numerical integrator [433]. The integration process begins with attitude motion for CS because its attitude motion does not depend on the servicer. A kinematic-driver computes the attitude of the SS, it points the SS camera towards the CS and use the relative position of the CS in the LVLH frame. Future upgrades may replace the kinematic-driver with dynamic integration of the SS attitude motion with active control torque law. In the following sections, we provide the governing equations for CS orbit and attitude motion as well as details of the SS pointing kinematic-driver. The software code for the dynamic simulation is provided in Appendix C.1.

B.1 Orbit Simulation

Coordinate systems for the orbital simulation is shown in Figure 1. The orbit frames \mathcal{F}_{co} and \mathcal{F}_{so} for CS and SS respectively are located in the Earth orbit about the J2000-ECI frame of reference. The Z axis of the orbital frames are nadir pointing while their X axis points in the direction of the orbital motion and Y complete the right hand. All vehicle motion in the ProxOps setting is relative to the \mathcal{F}_{co} frame which is the LVLH. The absolute orbital motion follows the standard two-body problem,

$$\ddot{\mathbf{r}} + \frac{\mu}{\|\mathbf{r}\|^3} \mathbf{r} = \mathbf{0}, \quad (1)$$

where $\mu = 3.986004418e14m^3s^{-2}$ is the Earth gravitational constant, and \mathbf{r} is the position of the spacecraft in the J2000-ECI frame. The orbital state is formulated as

$$\begin{bmatrix} \dot{\mathbf{r}} \\ \ddot{\mathbf{r}} \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \mathbf{1} \\ -\frac{\mu}{\|\mathbf{r}\|^3} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{r} \\ \dot{\mathbf{r}} \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{f} \end{bmatrix}, \quad (2)$$

$$\dot{\mathbf{x}}_o = \mathbf{A}_o \mathbf{x}_o + \mathbf{B}_o u_o, \quad (3)$$

where \mathbf{f} is the combined natural and artificial accelerations on the vehicle. The natural accelerations can be the result of planet geopotential, atmospheric drag, and Sun-Moon gravity *etc.*. Artificial accelerations may come from thruster models. For this investigation, the $\mathbf{B}_o u_o$ term is set to zero and delta-V is added directly to the orbit state. \mathbf{x}_o is the orbital state in J2000-ECI composed of the position and velocity of the spacecraft, $\mathbf{x}_o^T = [\mathbf{r}^T \ \dot{\mathbf{r}}^T]$. Orbit validation is performed using the CW formula for unperturbed orbital motion. The simulated unperturbed orbit provided in Eq. (3) is validated with the CW

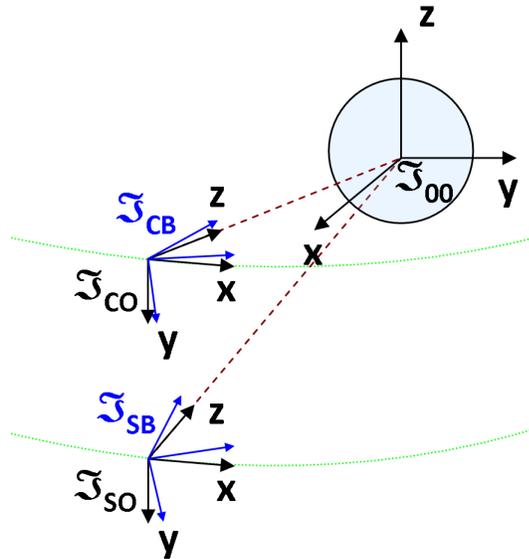


Figure 1: Coordinate system definitions. \mathcal{F}_{00} denotes J2000-ECI, \mathcal{F}_{co} denotes the location of the CS orbit and locates the CS body coordinate \mathcal{F}_{cb} , it is also the ProxOps local LVLH. \mathcal{F}_{so} denotes the SS orbit and locates the SS body coordinate \mathcal{F}_{sb} .

solution for close-proximity circular orbit. The CW Equation-of-Motion (EoM) for a body

in the Hill's frame is

$$\begin{aligned} \ddot{x} - 2\omega\dot{y} &= f_x \\ \ddot{y} + 2\omega\dot{x} - 3\omega^2y &= f_y \\ \ddot{z} + \omega^2z &= f_z, \end{aligned} \quad (4)$$

where ω is the Hill's frame orbital frequency such that $\omega^2r^3 = \mu$. The Hill's frame is defined as Y pointing to the zenith, X pointing aft, and Z complete the right-hand. The unperturbed solution to Eq. (4) expressed in LVLH is

$$\begin{bmatrix} \mathbf{r}_{co}^{co,so} \\ \dot{\mathbf{r}}_{co}^{co,so} \end{bmatrix}_t = \Phi(t) \begin{bmatrix} \mathbf{r}_{co}^{co,so} \\ \dot{\mathbf{r}}_{co}^{co,so} \end{bmatrix}_0 \quad (5)$$

where Φ is the *state transition matrix*,

$$\Phi(t) = \begin{bmatrix} 1 & 0 & 6[\omega t - \sin(\omega t)] & \frac{4}{\omega} \sin(\omega t) - 3t & 0 & \frac{2}{\omega} [1 - \cos(\omega t)] \\ 0 & \cos(\omega t) & 0 & 0 & \frac{\sin(\omega t)}{\omega} & 0 \\ 0 & 0 & 4 - 3\cos(\omega t) & \frac{2}{\omega} [\cos(\omega t) - 1] & 0 & \frac{\sin(\omega t)}{\omega} \\ 0 & 0 & 6\omega [1 - \cos(\omega t)] & 4\cos(\omega t) - 3 & 0 & 2\sin(\omega t) \\ 0 & -\omega \sin(\omega t) & 0 & 0 & \cos(\omega t) & 0 \\ 0 & 0 & 3\omega \sin(\omega t) & -2\sin(\omega t) & 0 & \cos(\omega t) \end{bmatrix}. \quad (6)$$

Equation (5) was used to validate the relative orbit difference between CS and SS computed by the absolute orbit integration. Figure 2 provides an example of the simulation validation using the unperturbed CW solution. Figure 2(a) shows match in orbit trajectory after three orbits. Figure 2(b) shows the SS trajectory in LVLH where delta-V adjustment at $Z = -20m$ is not available in the CW solution.

B.2 Attitude Simulation

The vehicle body frames \mathcal{F}_{cb} and \mathcal{F}_{sb} are located at the spacecraft CoM and is co-located with the respective orbital frames. The body frame is fixed to the vehicle, and its motion is governed by the Euler's equation for rigid body motion,

$$\mathbf{J}\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times \mathbf{J}\boldsymbol{\omega} = \boldsymbol{\tau}, \quad (7)$$

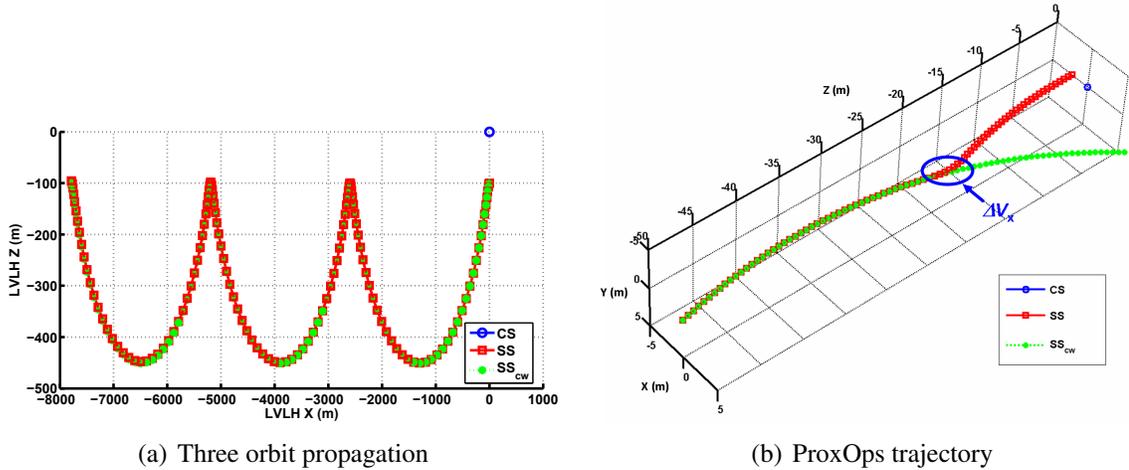


Figure 2: Example of orbit simulation validation expressed in the LVLH. In subfigure (b), the CW general solution does not make a course adjustment at $Z = -20m$ where the integrated solution takes into account the delta-V adjustment in the SS trajectory.

where \mathbf{J} is the vehicle MoI, $\boldsymbol{\omega}$ is the angular velocity, $\boldsymbol{\omega}^\times$ is the skew-symmetric cross product matrix associated with $\boldsymbol{\omega}$ and $\boldsymbol{\tau}$ is the externally applied torque from natural and artificial forces. The cross product matrix for any vector $\mathbf{a} = \begin{bmatrix} a_1 & a_2 & a_3 \end{bmatrix}^T$ is defined as

$$\mathbf{a}^\times = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix}. \quad (8)$$

An elegant method of transitioning the state attitude is by the use of Modified Rodriguez Parameters (MRP) [434]. First, the attitude of the vehicle is converted into MRP by applying a stereographic projection of the attitude quaternion that is denoted by $\mathbf{q} = \begin{bmatrix} \boldsymbol{\varepsilon}^T & \eta \end{bmatrix}^T$. $\boldsymbol{\varepsilon}$ is related to the axis angle ϕ and the attitude vector $\hat{\mathbf{a}}$ as $\boldsymbol{\varepsilon} = \hat{\mathbf{a}} \sin(\phi/2)$ and $\eta = \cos(\phi/2)$. The MRP is defined as,

$$\boldsymbol{\sigma} = \frac{\boldsymbol{\varepsilon}}{1 + \eta} = \hat{\mathbf{a}} \tan(\phi/4). \quad (9)$$

The angular velocity in MRP is derived as

$$\dot{\boldsymbol{\sigma}} = \mathbf{T}\boldsymbol{\omega}. \quad (10)$$

where \mathbf{T} is

$$\mathbf{T} = \frac{1}{4} \left[\left(\mathbf{1} - \boldsymbol{\sigma}^T \boldsymbol{\sigma} \right) \mathbf{1} + 2\boldsymbol{\sigma}^\times + 2\boldsymbol{\sigma} \boldsymbol{\sigma}^T \right]. \quad (11)$$

In Eq. (11), $\mathbf{1}$ is a 3×3 identity matrix and \mathbf{T} is a MRP-dependent transformation from Cartesian space to MRP space. Let define \mathbf{H} as a MRP-dependent MoI equivalent parameter in MRP space,

$$\mathbf{H} = \mathbf{T}^{-T} \mathbf{J} \mathbf{T}^{-1}, \quad (12)$$

and the transformation of body torques to MRP space is

$$\mathbf{u}_a = \mathbf{T}^{-T} \boldsymbol{\tau}. \quad (13)$$

Finally, let define a transformation matrix \mathbf{C} as

$$\mathbf{C} = -\mathbf{T}^{-T} \left[\mathbf{J} \mathbf{T}^{-1} \dot{\mathbf{T}} \mathbf{T}^{-1} + (\mathbf{J} \mathbf{T}^{-1} \dot{\boldsymbol{\sigma}})^\times \mathbf{T}^{-1} \right], \quad (14)$$

then the attitude state EoM can be formed as

$$\dot{\mathbf{x}}_a = \mathbf{A}_a \mathbf{x}_a + \mathbf{B}_a \mathbf{u}_a, \quad (15)$$

where \mathbf{x}_a is the attitude state in MRP space, $\mathbf{x}_a^T = \left[\boldsymbol{\sigma}^T \quad \dot{\boldsymbol{\sigma}}^T \right]$, and

$$\mathbf{A}_a = \begin{bmatrix} \mathbf{0} & \mathbf{1} \\ \mathbf{0} & -\mathbf{H}^{-1} \mathbf{C} \end{bmatrix}, \quad \mathbf{B}_a = \begin{bmatrix} \mathbf{0} \\ \mathbf{H}^{-1} \end{bmatrix}. \quad (16)$$

B.3 Target Pointing

The attitude of both CS and SS can be computed by numerically integrating Eq. (15) using the RK4 method [433]. A kinematic-driver is used to point the SS camera towards the CS instead of implementing an SS torque control law. Consider the coordinate frame specified in Fig. 3. The orientation of \mathcal{F}_{sb} can be developed using the position vector from \mathcal{F}_{sb} to \mathcal{F}_{cb} expressed in the LVLH (\mathcal{F}_{co}). This position vector is denoted by $\mathbf{r}_{co}^{sb,cb}$. The rotation matrix transforming a vector expressed in \mathcal{F}_{sb} to the LVLH is denoted by $\mathbf{R}^{co, sb}$. $\mathbf{R}^{co, sb}$

can be defined by three basis unit vectors expressed in LVLH,

$$\mathbf{R}^{co, sb} = \begin{bmatrix} \hat{\mathbf{x}} & \hat{\mathbf{y}} & \hat{\mathbf{z}} \end{bmatrix}. \quad (17)$$

Let the camera boresight axes align with \mathcal{F}_{sb} 's Z -axis. then the boresight unit vector $\hat{\mathbf{z}}$ is

$$\hat{\mathbf{z}} = \frac{\mathbf{r}_{co}^{sb, cb}}{\|\mathbf{r}_{co}^{sb, cb}\|}, \quad (18)$$

$\hat{\mathbf{x}}$ is chosen to be orthogonal to the LVLH Y -axis,

$$\hat{\mathbf{x}} = \frac{\begin{bmatrix} 0 & 1 & 0 \end{bmatrix}^T \times \hat{\mathbf{z}}}{\|\begin{bmatrix} 0 & 1 & 0 \end{bmatrix}^T \times \hat{\mathbf{z}}\|}. \quad (19)$$

Finally, $\hat{\mathbf{y}}$ is computed by crossing $\hat{\mathbf{z}}$ with $\hat{\mathbf{x}}$ to completing the right-handed coordinate system,

$$\hat{\mathbf{y}} = \frac{\hat{\mathbf{z}} \times \hat{\mathbf{x}}}{\|\hat{\mathbf{z}} \times \hat{\mathbf{x}}\|}. \quad (20)$$

\mathcal{F}_{sb} orientation relative to the LVLH in quaternion or Euler angles can be computed from the rotation matrix through standard conversions [435].

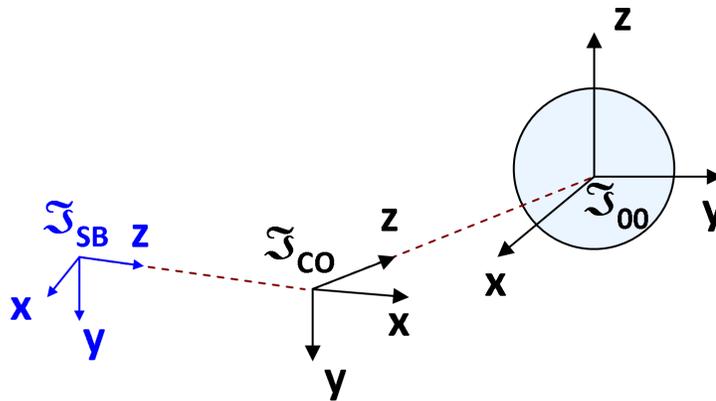


Figure 3: ProxOps local coordinate systems. Let the servicing CubeSat's camera coordinate to be the same as its body frame. LVLH is the client CubeSat's orbital frame \mathcal{F}_{co} \mathcal{F}_{00} is the J2000-ECI inertial coordinate system. The attitude kinematic-driver for the SS force its Z -axis to point in the direction of the LVLH.

Appendix C: Source Code

Source code is available as supplementary material software files and can also be downloaded from http://ai-automata.ca/01_research/2_code/phd_thesis_code.zip.

Copyright Notice

This software is being made available for individual research use only. You may use this work subject to the following conditions:

1. This work is provided ‘as is’ by the copyright holder, with absolutely no warranties of correctness, fitness, intellectual property ownership, or anything else whatsoever. You use the work entirely at your own risk. The copyright holder will not be liable for any legal damages whatsoever connected with the use of this work.
2. The copyright holder retain all copyright to the work. All copies of the work and all works derived from it must contain (1) this copyright notice, and (2) additional notices describing the content, dates and copyright holder of modifications or additions made to the work, if any, including distribution and use conditions and intellectual property claims. Derived works must be clearly distinguished from the original work, both by name and by the prominent inclusion of explicit descriptions of overlaps and differences.
3. The names and trademarks of the copyright holder may not be used in advertising or publicity related to this work without specific prior written permission.
4. In return for the free use of this work, you are requested, but not legally required, to do the following:
 - If you become aware of factors that may significantly affect other users of the work, for example major bugs or deficiencies or possible intellectual property issues, you are requested to report them to the copyright holder, if possible including redistributable fixes or workarounds.

- If you use the work in scientific research or as part of a larger software system, you are requested to cite the use in any related publications or technical documentation.
 - Shi, J.F. ‘Spacecraft Pose Estimation Using a Monocular Camera.’ Ph.D. Thesis, Dept. Mechanical and Aerospace Engineering, Carleton University, Ottawa, Canada, 2019.

This copyright notice must be retained with all copies of the software, including any modified or derived versions.

C.1: Dynamics Simulation Software

The orbit and attitude dynamic simulation are performed using MATLAB. The software used to generate the orbital data are provided as follows:

File Name:L5_test_cubeSat_orb.m
File Name:L3_t_cw.m
File Name:L2_t_cwrvphi.m
File Name:L2_t_cwphi.m
File Name:L2_t_cwdrag.m
File Name:L2_orb_thrstCtrl.m
File Name:L2_orb_soCir.m
File Name:L2_orb_kin.m
File Name:L2_ode_mrp_acc.m
File Name:L2_deep_rk4_att.m
File Name:L2_deep_rk4.m
File Name:L2_deep_Ag.m
File Name:L2_cnv_rv2egy.m
File Name:L2_cnv_rv2a.m
File Name:L2_cnv_re2v.m
File Name:L2_cnv_pv2vh_e.m
File Name:L2_cnv_pv2lvlh.m
File Name:L2_cnv_pv2classic2.m
File Name:L2_cnv_pv2classic.m
File Name:L2_cnv_pegy2e.m
File Name:L2_cnv_nu2EBH.m
File Name:L2_cnv_nM2t.m
File Name:L2_cnv_lvlh2pv_e.m
File Name:L2_cnv_hvec2ih.m
File Name:L2_cnv_h2p.m
File Name:L2_cnv_EBH2nu.m
File Name:L2_cnv_EBH2M.m

File Name:L2_cnv_a2nT.m
File Name:L2_att_kinDrvPnt.m
File Name:L1_t_vec_norm.m
File Name:L1_t_vec_dot.m
File Name:L1_t_vec_cross.m
File Name:L1_t_get_unitvec.m
File Name:L1_t_cross_vec.m
File Name:L1_t_cross_mat.m
File Name:L1_ode_mrp_pw2pd.m
File Name:L1_get_colmat.m
File Name:L1_cnv_tensorvec2mat.m
File Name:L1_cnv_R2q.m
File Name:L1_cnv_q2p.m
File Name:L1_cnv_p_ppd2w.m
File Name:L1_cnv_p_ppd2Fd.m
File Name:L1_cnv_p_pJwg2ABu.m
File Name:L1_cnv_p_p2F.m
File Name:L1_cnv_p_HC2AB.m
File Name:L1_cnv_p_gFti2u.m
File Name:L1_cnv_p_Fw2pd.m
File Name:L1_cnv_p_FJ2H.m
File Name:L1_cnv_p_FFdJpd2C.m
File Name:L1_cnv_pe2rmin.m
File Name:L1_cnv_pe2rmax.m
File Name:L1_cnv_p2q.m
File Name:L1_cnv_ap2ap.m
File Name:L1_chk_tol.m

C.2: Image Processing Software

The image processing software was developed in MATLAB. The software used to perform histogram equalisation, edge detection, Hough Transform and image point generation are provided as follows:

File Name:L5_test_img_bldsat.m

File Name:L5_test_cubeSat_pseEst.m

File Name:L5_test_cubeSat.m

File Name:L4_img_houghLnPlt.m

File Name:L4_fig_pltgeo.m

File Name:L2_img_rgb2gry.m

File Name:L2_img_houghPksM.m

File Name:L2_img_houghM.m

File Name:L2_img_houghLnsM.m

File Name:L2_img_houghEE.m

File Name:L2_img_harris_fst.m

File Name:L2_get_bstl_SunVecEci2.m

File Name:L2_geo_mkbp.m

File Name:L2_geo_GetTri2.m

File Name:L2_geo_GetRec2.m

File Name:L2_geo_GetRec.m

File Name:L2_geo_GetPts.m

File Name:L2_geo_GetPrmdN.m

File Name:L2_geo_GetPrmd3.m

File Name:L2_geo_GetElpsd.m

File Name:L2_geo_GetElps.m

File Name:L2_geo_GetCyln.m

File Name:L2_geo_Frm2Ctr.m

File Name:L2_geo_ellipsoid2Elp.m

File Name:L2_geo_ellipsoid2.m

File Name:L2_geo_ellipsoid.m

File Name:L2_geo_ellipse2.m
File Name:L2_geo_ellipse.m
File Name:L2_geo_cylindr3.m
File Name:L2_geo_cylindr2.m
File Name:L2_geo_bldsatt.m
File Name:L2_cnv_Fs2Fo.m
File Name:L2_cnv_Fob2ib.m
File Name:L2_cnv_Fio2ob.m
File Name:L2_cnv_cosb2iso.m
File Name:L2_astro_crd2crd.m
File Name:L1_get_rem.m
File Name:L1_cnv_xPQC2dtNC.m
File Name:L1_cnv_Rva2vb.m
File Name:L1_cnv_q2R.m
File Name:L1_cnv_q2o.m
File Name:L1_cnv_ioob2ib.m
File Name:L1_cnv_elp2rec.m

C.3: yBRIEF Software

The *y*BRIEF software was developed in MATLAB. The software used to compute *y*BRIEF performance comparison are provided as follows:

File Name:L5_test_abyss_latch.m
File Name:L4_xml_prseFleOcv.m
File Name:L4_xml_kp2fle.m
File Name:L4_xml_fle2mtch.m
File Name:L4_fig_ftreMtch.m
File Name:L2_xml_mat2fle.m
File Name:L2_xml_fleHdr.m
File Name:L2_xml_fleEdr.m
File Name:L2_img_smthImgPtch.m
File Name:L2_img_sift_key2dat.m
File Name:L2_img_sift_im2pgm.m
File Name:L2_img_sift_exe.m
File Name:L2_img_pyrrnd_orb.m
File Name:L2_img_orbOct.m
File Name:L2_img_orb.m
File Name:L2_img_ocv_key2dat.m
File Name:L2_img_ocv_exe.m
File Name:L2_img_LUT_gen.m
File Name:L2_img_LUT_fnd.m
File Name:L2_img_lmtKP.m
File Name:L2_img_kpMtch.m
File Name:L2_img_kp2scl.m
File Name:L2_img_knn_exe.m
File Name:L2_img_kNNocv.m
File Name:L2_img_kNNOct2.m
File Name:L2_img_kNNbin2.m
File Name:L2_img_kNNang2.m

File Name:L2_img_hrsPymd.m
File Name:L2_img_histeq.m
File Name:L2_img_harris_fst.m
File Name:L2_img_cnrAng.m
File Name:L2_img_abyss2.m
File Name:L2_dr_qryRecSimChkTol.m
File Name:L2_dr_qryEqu.m
File Name:L2_dr_qryCosSimChkTol.m
File Name:L2_dr_mtcROC.m
File Name:L2_dr_kNNqry.m
File Name:L2_dr_getkmn.m
File Name:L2_dr_cmpSmCosBin.m
File Name:L2_dr_cmpROC.m
File Name:L2_dr_cmpRec.m
File Name:L2_dr_3qry2DChkRec.m
File Name:L2_dr_3qry2DChkMdxRec.m
File Name:L2_dr_3qry2DChkMdx.m
File Name:L2_dr_3qry2DChk.m
File Name:L2_dr_2qry2DChkRec.m
File Name:L2_dr_2qry2DChkMdxRec.m
File Name:L2_dr_2qry2DChkMdx.m
File Name:L2_dr_2qry2DChk.m
File Name:L1_xml_getRoot.m
File Name:L1_xml_getMat.m
File Name:L1_xml_getDat.m
File Name:L1_xml_getBrnch.m
File Name:L1_xml_getAtrb.m
File Name:L1_t_unormMat.m
File Name:L1_t_srtRowUnq.m
File Name:L1_t_srtRow.m
File Name:L1_t_srtMat.m

File Name:L1_t_srtCol.m
File Name:L1_t_rnd.m
File Name:L1_t_recBnd.m
File Name:L1_t_randDstb.m
File Name:L1_t_randCol.m
File Name:L1_t_mtchMrg.m
File Name:L1_t_matSrtUnq.m
File Name:L1_t_l2Nrmqry.m
File Name:L1_t_Kunnorm.m
File Name:L1_t_Knorm.m
File Name:L1_t_knkrRt.m
File Name:L1_t_knkrR.m
File Name:L1_t_knkrP.m
File Name:L1_t_knkr.m
File Name:L1_t_K1Dnorm.m
File Name:L1_t_hamDistqry.m
File Name:L1_t_getEven.m
File Name:L1_get_norm.m
File Name:L1_cnv_rec2pol_rad.m
File Name:L1_cnv_num2bit.m
File Name:L1_cnv_idx2idx.m
File Name:L1_cnv_H2ab_fst.m
File Name:L1_cnv_bte2bit.m
File Name:L1_cnv_bit2bte.m
File Name:L1_cnv_bdang2pi_fst.m
File Name:L1_cnv_ary2mat.m

C.4: BoVW Software

The BoVW analysis was performed in MATLAB. The BoVW software are provided as follows:

File Name: L5_test_dr_bovw_0_end2end.m

File Name: L4_ptn_bbxGt.m

File Name: L4_plt_rec.m

File Name: L4_fig_addfrm.m

File Name: L2_ptn_QuadFn.m

File Name: L2_ptn_knnFst.m

File Name: L2_ptn_fndPtch.m

File Name: L2_ptn_ftClstrfst.m

File Name: L2_ptn_ftClstr.m

File Name: L2_ptn_dicLbl.m

File Name: L2_ptn_clsWinf.m

File Name: L2_ptn_clsNBinf.m

File Name: L2_ptn_bbxIoU.m

File Name: L2_img_rgb2gryPca.m

File Name: L2_img_read2bw.m

File Name: L2_img_pca_g.m

File Name: L2_img_pca.m

File Name: L2_img_mkfltrS.m

File Name: L2_img_kMeanClstr.m

File Name: L2_img_histeqC.m

File Name: L2_img_ftSMR8Bnk.m

File Name: L2_img_ftRspMap.m

File Name: L2_img_ftrMR8Fst.m

File Name: L2_img_ftrMR.m

File Name: L2_img_ftNrm.m

File Name: L2_img_flpY.m

File Name: L2_img_dr_sift.m

File Name: L2_img_cdeBkPtch.m
File Name: L2_img_cdbkClstr.m
File Name: L2_img_bbx2bbx.m
File Name: L2_dr_pcai.m
File Name: L2_dr_pca.m
File Name: L2_dr_hstMdlFtrVcxInf.m
File Name: L2_dr_hstMdlFtrVcx.m
File Name: L2_dr_binMap.m
File Name: L1_t_vecErr.m
File Name: L1_t_unormMat.m
File Name: L1_t_svds.m
File Name: L1_t_stmp3.m
File Name: L1_t_pNorm.m
File Name: L1_t_normMatL1.m
File Name: L1_t_normMat.m
File Name: L1_t_dtIni.m
File Name: L1_t_dt.m
File Name: L1_cnv_xPQC2dtNC.m
File Name: L1_cnv_xPQC2dhNC.m
File Name: L1_cnv_fPQC2fNC.m
File Name: L1_cnv_dtNC2xPQC.m
File Name: DAT_constants_ptn.m

C.5: PCA appearance-based Software

The PCA appearance-based pose estimation analysis was performed in MATLAB. The PCA appearance-based software are provided as follows:

File Name:L5_test_pca.m

File Name:L5_test_epca_2.m

File Name:L5_test_epca_1.m

File Name:L3_img_pca_tst.m

File Name:L2_img_pca_tst.m

File Name:L2_img_pca_trn.m

File Name:L2_img_pca_g.m

File Name:L2_img_pca_eulTst.m

File Name:L2_img_pca_eulTrn.m

File Name:L2_img_pca.m

File Name:L1_t_pNrm.m

File Name:L1_get_colmat.m

File Name:L1_f_ScanChar.m

File Name:L1_f_GetChar.m

File Name:L1_cnv_RCM2NMh.m

File Name:L1_cnv_mat2vech.m

File Name:L1_cnv_mat2ary.m

C.6: SoftPOSIT Software

The *SoftPOSIT* PnP pose estimation analysis was performed in MATLAB. The *SoftPOSIT* software are provided as follows:

File Name:L2_img_sPST_xM.m

File Name:L2_img_sPST_wM.m

File Name:L2_img_sPST_simpPrm.m

File Name:L2_img_sPST_RxytTzi.m

File Name:L2_img_sPST_RSH2.m

File Name:L2_img_sPST_Qfi.m

File Name:L2_img_sPST_nMatch.m

File Name:L2_img_sPST_mxPosRat.m

File Name:L2_img_sPST_lstBta.m

File Name:L2_img_sPST_lean.m

File Name:L2_img_sPST_ip2vw.m

File Name:L2_img_sPST_iniBta.m

File Name:L2_img_sPST_hM.m

File Name:L2_img_sPST_Fnb3.m

File Name:L2_img_sPST_Fnb2.m

File Name:L2_img_sPST_Fnb.m

File Name:L2_img_sPST_Fn.m

File Name:L2_img_sPST_dltMxMnD.m

File Name:L2_img_sPST_D34.m

File Name:L2_img_sPST_d2Mat.m

File Name:L2_img_sPST_D2.m

File Name:L2_img_sPST_cntrdMtch.m

File Name:L2_img_sPST_cntrdBta.m

File Name:L2_img_sPST_btaTrD.m

File Name:L2_img_sPST_bp1.m

File Name:L2_img_sPST_aMatIni.m

File Name:L2_img_sPST_A2r.m

File Name:L2_img_recCul.m
File Name:L2_img_pj3d22dc.m
File Name:L2_img_pj3d22d.m
File Name:L2_img_GenOctOdr.m
File Name:L2_img_GenOct.m
File Name:L2_img_extrcInfo.m
File Name:L2_img_cylFndOri.m
File Name:L2_img_cylCul.m
File Name:L2_img_cul3DbkPts.m
File Name:L2_img_3d22dvw.m
File Name:L1_t_newton_1DME.m
File Name:L1_t_get_alph.m
File Name:L1_t_getInvsel.m
File Name:L1_t_DltPosOri.m
File Name:L1_t_DltPos.m
File Name:L1_get_rss.m
File Name:L1_get_rowmat3.m
File Name:L1_get_centroid.m
File Name:L1_fig_axsMat.m
File Name:L1_cnv_R2o_rad.m
File Name:L1_cnv_R2o.m
File Name:L1_cnv_R2aphi.m
File Name:L1_cnv_o2R_rad.m
File Name:L1_cnv_o2R.m
File Name:L1_cnv_bdangnpippi.m

C.7: *ePnP* Software

The *ePnP* pose estimation analysis was performed in MATLAB. The *ePnP* software are provided as follows:

File Name:L5_test_pnp_endToEnd.m

File Name:L5_test_img_epnp.m

File Name:L3_img_epnp_resetMxNumCspd.m

File Name:L3_img_epnp_mxNumCspd.m

File Name:L3_img_epnp_intrsc.m

File Name:L3_img_epnp_ini.m

File Name:L3_img_epnp.m

File Name:L2_img_rsc_Hmg.m

File Name:L2_img_rnd3dPnt.m

File Name:L2_img_relErr.m

File Name:L2_img_Pwc2Rt.m

File Name:L2_img_prjNseV.m

File Name:L2_img_prjNse.m

File Name:L2_img_pixDif.m

File Name:L2_img_pix2pix.m

File Name:L2_img_loc2km.m

File Name:L2_img_kNNOct.m

File Name:L2_img_kNNbin.m

File Name:L2_img_kNNang.m

File Name:L2_img_iPQ2pgm.m

File Name:L2_img_hmgCrect.m

File Name:L2_img_getRePrj.m

File Name:L2_img_flpX.m

File Name:L2_img_epnp_rndPse.m

File Name:L2_img_epnp_gnItr.m

File Name:L2_img_epnp_getRho.m

File Name:L2_img_epnp_getMnErr.m

File Name:L2_img_epnp_getL610.m
File Name:L2_img_epnp_getBta.m
File Name:L2_img_epnp_genM.m
File Name:L2_img_epnp_ctlPts.m
File Name:L2_img_epnp_cc34.m
File Name:L2_img_epnp_c2p.m
File Name:L2_img_epnp_bcCrd.m
File Name:L2_img_epnp_b1N.m
File Name:L2_img_epnp.m
File Name:L2_img_abyss_polCnvLfstNDE.m
File Name:L2_img_abyss_ltchOct.m
File Name:L1_t_triDistCos.m
File Name:L1_t_triAngCos.m
File Name:L1_t_svdAx0.m
File Name:L1_t_srtUnq.m
File Name:L1_t_rsc_stdDev.m
File Name:L1_t_rsc_selVal.m
File Name:L1_t_rsc_p2T.m
File Name:L1_t_rsc_getT.m
File Name:L1_t_pop.m
File Name:L1_t_knkrPt.m
File Name:L1_t_hamDist.m
File Name:L1_t_getNonUnq.m
File Name:L1_t_diag.m
File Name:L1_t_angSub.m
File Name:L1_t_aNbM2HNM2.m
File Name:L1_t_aNbM2HNM.m
File Name:L1_get_diag.m
File Name:L1_get_allMin.m
File Name:L1_get_allMax.m
File Name:L1_cnv_R2o.m

File Name:L1_cnv_pol2rec_rad.m

File Name:L1_cnv_o2R_yry_rad.m

Publications

1. Shi, J.F., Ulrich, S., and Ruel, S., An unsupervised method of infrared spacecraft image foreground extraction. *AIAA Journal of Spacecraft and Rockets*, (under review), 2019.
2. Shi, J.F., Ulrich, S., and Ruel, S., Saliency detection and 6-DOF pose estimation of monochromatic monocular spacecraft images. *IEEE Trans. on Image Processing*, (under review), 2019.
3. Shi, J.F., Ulrich, S., and Ruel, S., Real-time saliency detection for grayscale and colour images. *Intl. Journal of Computer Vision*, (under review), 2019.
4. Shi, J.F., Ulrich, S., and Ruel, S., Regional method for monocular infrared image spacecraft pose estimation. *Proc. of the AIAA Space Conf. and Exhibit*, Orlando, FL, September 17-19, 2018.
5. Shi, J.F., Ulrich, S., and Ruel, S., Level-set and image statistics for pose estimation of satellites. *Proc. of the Intl. Conf. on Control, Dynamic Systems, and Robotics*, Niagara Falls, Canada, June 7-9, 2018.
6. Shi, J.F., Ulrich, S., and Ruel, S., CubeSat simulation and detection using monocular camera images and convolutional neural networks. *Proc. of the AIAA Guidance, Navigation, and Controls Conf. and Exhibit*, Kissimmee, FL, January 8-12, 2018.
7. Shi, J.F., Ulrich, S., and Ruel, S., Spacecraft component recognition using a code-book of texture images. *Proc. of the AIAA Space Conf. and Exhibit*, Orlando, FL, September 12-14, 2017.
8. Shi, J.F., Ulrich, S., and Ruel, S., International space station image extraction from a dynamic environment using deep learning. *Proc. of the Intl. Conf. on Control, Dynamic Systems, and Robotics*, Toronto, Canada, August 21-23, 2017.

9. Shi, J.F., Ulrich, S., and Ruel, S., yBRIEF: a study of non-Gaussian binary elementary features. *Proc. of the IEEE Intl. Conf. on Image, Vision and Computing*, Chengdu, China, June 2-4, 2017.
10. Shi, J.F., Ulrich, S., and Ruel, S., A comparison of feature descriptors using monocular thermal camera images. *Proc. of the IEEE Intl. Conf. on Control, Automation and Robotics*, Nagoya, Japan, April 22-24, 2017.
11. Shi, J.F., Ulrich, S., and Ruel, S., Spacecraft pose estimation using principal component analysis and a monocular camera. *Proc. of the AIAA Guidance, Navigation, and Controls Conf. and Exhibit*, Grapevine, Texas, Jan 9-13, 2017.
12. Shi, J.F., Ulrich, S., and Ruel, S., Spacecraft pose estimation using a monocular camera. *Proc. of the Intl. Astronautical Congress of Guidance, Control, and Dynamics*, Guadalajara, Mexico, Sept 26-30, 2016.
13. Shi, J.F., Ulrich, S., Chamitoff, G.E., Morrell, B.J., and Allen A., Trajectory optimization for proximity operations around tumbling geometrical constraints via Legendre polynomials. *Proc. of the AIAA Astrodynamics Conf. and Exhibit*, Long Beach, CA, Sept 12-15, 2016.
14. Shi, J.F., and Ulrich, S. SoftPOSIT enhancements for monocular camera spacecraft pose estimation. *Proc. of the IEEE Intl. Conf. on Methods and Models in Automation Robotics*, Miedzyzdroje, Poland, Aug 28-30, 2016.
15. Shi, J.F., Ulrich, S., and Ruel, S., Uncooperative spacecraft pose estimation using an infrared camera during proximity operations. *Proc. of the AIAA Space Conf. and Exhibit*, Pasadena, CA, Aug 31-Sept 2, 2015.
16. Shi, J.F., Ulrich, S., and Allen, A., Optimal trajectory guidance for spacecraft robotic servicing missions. *Symp. on Advanced Space Technologies in Robotics and Automation*, Noordwijk, The Netherlands, May 11-13, 2015.
17. Shi, J.F., Ulrich, S., and Ruel, S., Spacecraft adaptive attitude control with application to space station free-flyer robotic capture. *Proc. of the AIAA Guidance, Navigation, and Control Conf.*, Kissimmee, FL, Jan 5-9, 2015.

18. Shi, J.F., and Ulrich, S., A direct adaptive control law using modified rodrigues parameters for ISS attitude regulation during Free-Flyer capture operations *Proc. of the Intl. Astronautical Congress Journal of Guidance, Control, and Dynamics*, Toronto, Canada, Sept 29-Oct 3, 2014.

Biographical Sketch

Jian-Feng Shi received his BEng with high distinction in Aerospace Engineering from Carleton University (Ottawa, Canada) in 2001, and MASc degree in Aerospace Engineering from University of Toronto (Toronto, Canada) in 2004. From 2001 to 2003 he was the Deputy Systems Engineering lead at Orbital Technology Corporation for the International Space Station (ISS) Plant Research Unit project. Jian-Feng was also a Guidance, Navigation, and Control (GNC) Engineer at MacDonald Dettwiler and Associates Ltd. (MDA) from 2004 to 2015. While at MDA he worked on the ISS Mobile Servicing System control software design and mission analysis, the Next Generation Canadarm design, and the Space Infrastructure Servicing program mission planning and software tools development. Jian-Feng was the robotic analysis lead for the ISS robotic capture of the JAXA HTV, SpaceX Dragon, and Orbital Sciences Cygnus free-flyer vehicles. From 2003 to 2017, Jian-Feng has instructed at the University of Toronto Da-vinci Engineering Enrichment Program teaching secondary-school students in spacecraft orbital mechanics, robotics, computer vision, and project management.

Jian-Feng is the recipient of many awards, including the Canadian Space Agency Certificate of Distinction for Professionalism and Excellence, NASA Achievement Award, the Alexander Graham Bell Canada Graduate Scholarship from the Natural Sciences and Engineering Research Council of Canada, the University of Toronto Arbor Award, and the prestigious Engineers Canada-Manulife Scholarship. In 2007, he was awarded a fellowship to attend the Space Studies Program of the International Space University held at the Beijing University of Aeronautics and Astronautics, in China.

Jian-Feng's research interest is in spacecraft orbit, attitude and robotic dynamic modeling, advanced vehicle GNC, machine learning and deep learning techniques on object localisation and segmentation, and computer vision techniques on image features, image saliency, and spacecraft pose estimation. Jian-Feng Shi is a registered Professional Engineer in the province of Ontario and a Project Management Professional. He is a senior member of the AIAA and is the vice-Chair on the AIAA Space Automation and Robotics Technical Committee.