Vertex-based Facial Animation Transfer System for Models with Different Topology; A Tool for Maya

By

Vatsla Chauhan

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

Master of Computer Science

In

Department of Computer Science

Carleton University
Ottawa, Ontario

Abstract

The transfer of realistic facial animations between two 3D models is limited and/or too cumbersome for general use. In this paper, we target transfer of facial expression animation and facial animation sequences within the 3D software modelling tool Autodesk Maya to eliminate some of the issues that animators come across while creating animations, along with making their process much faster and efficient. While current animation transferring processes in Maya work well, the downfall of these methods is that they only work properly if the source and target models have same vertex count, vertex order, and topology; in other words, the model has to be the same, and there is a lot of manual work required beforehand. We propose a system that eliminates these issues in order to achieve smooth and realistic animations while maintaining a 3D model-agnostic focus; the system features facial animation transfer between different models, i.e. different topology, vertex ordering and vertex count. The system is purely vertex-based; this helps to speed up the process since no rigging is necessary. Furthermore, the system reduces the amount of manual effort in order to accomplish a smooth and realistic facial animation.

Acknowledgements

I would like to thank my two co-supervisors, Professor Chris Joslin and Professor Audrey Girouard, for their assistance during this research. Their advices and suggestions have consistently helped to steer this work in a productive direction and have greatly contributed to the quality of this thesis. I also want to thank Eduardo Soto for his unconditional support.

Table of Contents

List of Tables

This is the List of Tables.

List of Figures

This is the List of Figures.

List of Appendices

This page lists all the appendices.

Abbreviations

ICP – Iterative Closest Point
FACS – Facial Action Coding System
RBF – Radial basis Function
LBS – Linear Blend Skinning
AU – Action Unit
kCCA – kernel Canonical Correlation Analysis
CCA – Canonical Correlation Analysis
FRGC – Face Recognition Grand Challenge
AVO – Audio-Visual Objects
FAPS – Face Animation Parameters
k-NN – k-Nearest Neighbour
SDR – Software Defined Radio
NET – Neighbour-Expression Transfer
MU – Motion Unit
MUP – Motion Unit Parameters
DPCM – Differential Pulse Code Modulation
DCT – Discrete Cosine Transform
BIFS – Binary Format for Scenes
FFD – Free Form Deformation

Chapter 1: Introduction

Facial Animation is a serious and ongoing challenge for the Computer Graphic industry. This provides a way for 3D face models to portray emotions. Improving the production speed of the animation process remains an unsolved problem, while maintaining the integrity of the animation. Nowadays, experienced animators and modellers create each facial expression animation to ensure that the animation achieves best quality, but this process is time-consuming, tedious, and requires much effort.

This thesis presents a facial expression animation transfer system. It automatically transfers the facial animation created from one 3D face model to another with minimal manual setup. It ensures that animators and modellers receive an animation that is highly realistic and very smooth using a combination of linear and non-linear deformation transfers. The face model can depict complex expressions easily thanks to a vertex matching algorithm. As a result, we dramatically reduce the production time for the creation of facial expression animations for the entertainment industry. Based on our facial mesh deformation method, our system allows usability of facial animation transfer for both experienced and inexperienced users. We researched and documented previous/traditional methods used in the field for facial animation transfer, to develop a solution to overcome this problem.

Our work differs from previous facial expression animation transfer techniques, because previous work was oriented towards transferring animations from videos or images, while ours focuses on transferring from one face model to another. In addition, it also overcomes a few issues that similar transferring techniques between face models have come across.

We used face models from TurboSquid library online. Our system is: generic (the facial models can have any type of topology and/or vertex count), flexible (the system can be easily extended to include facial model with different types of deformation methods), independent (the face model can be any shape and appearance), and reusable (the source face model is reused for many target face models, and vice versa).

Synthesis and retargeting of facial expressions require manual work to achieve realistic expressions due to difficulties faced in capturing high-quality dynamic expression data (Wang et.al. [24]). Through initial research, it was determined that the proposed system is viable, and therefore we focused on developing an application for the automatic facial expression animation transfer system. Facial animation introduces many obstacles that hinder the usability process of the method like time, cost, effort and complexity. Our system gives the user freedom to use any 3D face model to transfer the facial animation, regardless of their topology. To obtain a desired result for realistic facial animations, the traditional animation pipelines requires individual animated face model by hand, which is a time-consuming and tiresome process.

1.1 Problem Statement

While creating animations, animators would like to reuse good animations from one model onto another because the process of creating animations is time-consuming, tedious, and costly. There are a few existing tools that allow the animators to reuse animations, one of which is called Blendshapes (a deforming tool for creating animations in Maya), which allows the animators to reuse animations between similar models, i.e., same vertex count, vertex order, and topology. This means that reusability between different models is restricted to these constraints. This serves as a motivation to development of proposed system. The proposed system would be able to reuse/transfer facial animations between similar and different models, i.e., same or different vertex count, vertex order, and/or topology. The resultant facial animation would maintain the shape and appearance of the target face model as well as adapting the facial animation transfer from the source face model.

1.2 Our Solution

We propose a system that performs the transfer of 3D facial expressions and animations from custom source facial model to any target face models. The system includes a source 3D face model which has professional facial expression animations, and a 3D target model which doesn't have any animations associated to it. Then, the system transfers the facial expression animation from the source face model to the target face model of different topology, vertex count, and/or vertex order automatically. This method assumes no initial constraints other than a 3D source face model with facial animations, and a plain 3D target face model. We provide two different solutions, the first one is Weighted Cluster Method which is a simple, fast, and smooth method to transfer animations, and the second one is Vertex-Based Animation Transfer method which learns from the first method and uses more complex algorithms to preserve deformation data better than the first method. Designed for experienced and inexperienced modellers and animators, this solution aims to tackle the issue of facial expression and animation transfer in an efficient, reusable, and modifiable way.

1.3 Contribution

The current facial animation transfer techniques are time-consuming and require a lot of effort as they rely on traditional manual creation of the facial animations. A major advantage of our system is that the system allows reusability of the facial animation for different face models. This system creates a smooth and realistic 3D facial expression animation.

The key contributions of this thesis to the field of computer graphics and entertainment industry are:

- **Two facial mesh deformation method**: consists of landmark selection, vertex matching algorithm, deformation of animation for source and target models.
- **Pose-to-pose expression transfer method**: constitutes a variety of poses from the professionally animated source model onto any target model, independent of its shape and topology. Easy and ready to use on any target model without the need of a base model in the form of an Autodesk Maya Python script.
- **Animation Sequence transfer method**: replicates animation sequences from any base model to a target model. This further extends the pose-to-pose expression transfer by making use of an entire animation with various poses.

1.4 Outline of Thesis

The present document is organized as follows:

Chapter 2 describes the previous research on facial retargeting. It also describes the different approaches involved in the creation of facial expression animation using a variety of parameters. The review of weight alignment and proportional editing is also described in depth. It provides a better understanding on MPEG-4 FAPS, and how to incorporate the technique into practice with the system. Finally, it describes in depth the previous work in facial animation transfer using different sources like video, image, and 3D face models.

Chapter 3 describes the overall proposed system along with in-depth explanation of each component. This includes transformation-based animation extraction, landmark definition and creation of control points, and transfer procedure for pose-to-pose as well as sequence facial expression animation transfer.

Chapter 4 shows the results obtained from the system. It contains a set of testimonies from animators/modellers about creation of facial animations. A few advantages about the method and a few limitations that must be investigate in the future.

Chapter 5 concludes the document and discusses potential future research directions.

Chapter 2: State of the Art

In this section, we provide some related work that has led up to this point. This includes a description of contributions made in the form of facial retargeting, facial expression animation, weight alignment and proportional editing, MPEG-4, and the transfer of expressions. While this paper focuses on the smooth automatic transfer of facial expressions, previous work done in related areas mentioned below are crucial to the informed decisions and design of this system.

2.1 Facial expression retargeting

Facial animation can be performed with similar face assets without retargeting, but when performing with different face assets, it will show unwanted results. That's where retargeting comes in. Facial retargeting is a feature that allows retargeting facial expressions on a character's face, which more specifically means translating and fitting the expression animation between two separate and possibly different faces. In other words, character A's animation run cycle would be reused on another character of similar or different proportions. The original (source) face provides the animation data to be retargeted, and the target face is where you would retarget the animation. The most commonly used tools to achieve retargeting are MotionBuilder, Maya, WebAnimate, and many more.

Chuang and Bregler [1] developed a flexible 3D system that would create facial animation using a combination of motion capture data and blendshape interpolation. The method provides the motion capture process to drive the facial animation rather than animators animating by hand. This method is effective with different shapes in original and target actors. The computer vision techniques track the facial features of the user in the video, and the system automatically makes the sets of the key-shaped to model characteristic motion variations. Then the facial data is "decomposed into a weighted combination of the key-shape set [1]". Finally, the target key-shapes are made for each animated face model. It is important to keep the key shapes consistent because if the extremeness of the key-shaped is contradicting, then the output sequence will have undesired results. For example, for a video with 386 frames recorded at 30 frames per second, they used 10 key-shapes for the eye area, and 12 key-shapes for the mouth area. The issue with this system is that the assumption of linear interpolation is limited, and a better method could be found for the extraction of the key-shapes and then the key weights could exist. The resultant face model has the same facial motion as the original one, and the user has full control over the new face's appearance.

A real-time 3D facial tracking system was developed by Chai et.al. [2] that would be able to create a set of realistic facial expressions from a pre-processed motion capture database, and the user would be able to control all the expressions. As the recorded data are from a video, they would have noise, low-resolution, and certain errors. The system would translate all these low-quality signals into high-quality 3D facial expressions. They developed a retargeting technique "whose run-time computation is constant independent of the complexity of the model [2]". The technique would adapt to the synthesized motion in order to animate all the vertices of a model at run-time. The system can track 6 DOFs like a yaw, pitch, roll, and 3D position. The system automatically extracts 15 control parameters [mouth (6), nose (2), eye (2), and eyebrow (5)] that describe the facial expressions of the model. The efficient retargeting technique comprises of four stages: motion vector interpolation, dense surface correspondences, motion vector transfer, and target motion synthesis. The two major issues with the system are that it sometimes loses lip details, and speech is not an input for the model. The system doesn't have enough user study to evaluate efficiency and quality. The expressions on the resultant model might not be as realistic as the original model because the synthesized motion is interpolated and adapted from the motion capture data.

Along the same lines, Curio et.al. [3] developed a 3D realistic facial animation system that would have the ability to decompose facial motion capture data into semantically meaningful motion channels. This is possible by using the reference of Facial Action Coding Units. Due to the combination of the high spatial resolution of a 3D scanner and high temporal accuracy of motion capture data, the resultant 3D face model is very realistic with sparse measurements. The system would use a verbal description of action units to guide actors to perform facial actions. These actions are recorded in two ways: 3D scanner and optical motion capture system. The transfer is possible because of the semantic match of these two ways. With the help of ABW scanner, the shape and colour data for the face model was captured. Any existing rigid head motion is removed using the Iterative Closest Point (ICP) algorithm. Then the neutral human face is captured using 3D software called heads CySlice, where the network is converted into a triangle mesh with 3980 vertices. Finally, the facial shapes are loaded in Autodesk 3ds Max for each face within a single morph object. The system showed that method A (3D scanning) was preferred as compared to method B (Motion Capture) by the help of one-tailed t-test and

ANOVA test. The system could also automate the very time-consuming manual Facial Action Coding System (FACS) [14] annotation procedure.

Dutreve et.al. [4] developed a 3D system that would transfer facial animation in real-time. The input for the system can either be an existing 3D animation or 2D data proving by a video tracker or a motion capture system. The system would train an Radial Basis Function (RBF) network along with providing a geometric transformation between the original face model and the target face model. The system deforms the GPU's target mesh with Linear Blend Skinning (LBS). They proposed a procedural method called a quasi-automatic approach which computes for each vertex an influence weight to a feature point as per the distance. This method reduces the manual work to positioning feature points on the target model. "The method is based on the weight function from the ellipsoid function which allows it to define an independent parameter for each Action Unit (AU) since the feature points influence area is not regular [4]". For each vertex, the method compares similar weights of the feature points and picks out maximum weights to normalize them to have a sum weight of 1. Thus, it provides automatic rigging. They also developed a procedural technique that would be able to automatically rig the target face by generating vertices weights for the deformation process. The animator can choose which feature they want to transfer like only eyes or nose. This can be done by manually selecting feature points, and when you select a new feature point in the original face model, the target face model will adapt to it as per the expressions of the original face model and the morphology of the target face. The system uses 21 feature points: 3 for each eyebrow, 2 for each eye, 3 for the nose, and 8 around the mouth. To test the efficiency of the system, they developed a testing application based on Pyramidal Lucas

Kanade Feature Tracker [35]. The issues that they would work on in future are a full-automatic detection of feature points on 3D virtual faces, to improve the quality of our automatic rigging, remove the need of marker in video tracker, and ability to capture and transfer fine details like wrinkles.

Song et.al. [5] developed a robust retargeting system that represents the characteristics of the target face model by using their hybrid method based on Radial Basis Function (RBF) and kernel Canonical Correlation Analysis (kCCA). Regardless of its linearity of the sample data pairs, these methods can transfer the source animation to target animation. They use blendshapes of human face models suitable for retargeting, instead of using motion capture markers directly as input data. In similar existing techniques, the system often fails due to the complexity of the cross-mapping, but in this process; this issue is eliminated using blended linear and non-linear retargeted models. They created a hybrid of RBF and kCCA because RBF understands and evaluates the training data efficiently, and kCCA outperforms for dense correspondence, thus avoiding over fitting artefacts. So, by combining the two, the system will be able to cope with changing source inputs and evaluate visually appealing and characteristic facial shapes. “RBF is one of the most powerful interpolation methods to learn the best weight values between source and target data pairs [5]”. Although RBF cannot outperform for non-linear data, when an animator makes expression pairs and if two source expressions are similar, they are mapped to different targets, it will cause conflict in the training set. “Canonical Correlation Analysis (CCA) is a statistical analysis method that is used for defining the correlation between two sets of multivariate data. kCCA is a nonlinear version of CCA that uses kernel function [5]”. The kCCA method is used to find the base pair while maximizing the correlation between reduced input and output data. The system uses the Laplacian motion warping to enhance the characteristics of the retargeted face model while keeping the fine details of the source face model. Now the system can encode the key details from the original face model, and this allows the creation of subset key values in the target interval and original retargeted motion. After testing the system with morphologically different faces along with human faces, the system has proven to be computation efficient and robust.

With recent advances in facial expression transfer, the acquiring of high-fidelity 3D facial performance data with the high spatial-temporal resolution is possible, but the transfer is often limited to large-scale facial deformation. Xu et.al. [6] proposed a technique which would decouple the high-fidelity facial performance data into high-level facial feature lines, large-scale deformation, and fine-scale motion details. It will be transferred to reconstruct the retargeted face model animation in an optimized and efficient manner. The system takes a source reference expression, a source facial animation sequence and a target reference expression as input, combines the blendshape representation with deformation gradient technique and produces a retargeted animation sequence. They created a transfer process that would construct a target face from the corresponding source face base as people are highly sensitive to the movement of eye and mouth on the face. The system will calculate the approximate matching of the deformation between feature lines of target face and source face while setting the weight to 100. In order to optimize the process, they created an optimization algorithm which consists of face bases and weight vectors of the target sequence. The system will give users the ability to modify and control the retargeted face sequences in the spatial-temporal domain. This system combines geometric mapping and blendshape mapping as geometric mapping helps to transfer the facial performance to the target face but does not allow to control or modify the transferred performance. Blendshape mapping allows you to modify the retargeted animation.

Existing animation methods either created matching facial expressions of actor and target character manually or used characters with similar facial proportions. To address this issue, Ribera et.al. [7] developed an approach that would be able to automatically retarget the facial expressions from actor to target character. This approach shows that with a given training set of actor's range of motion, it is possible to create an accurate parallel parameterization, in an unsupervised manner with differences in the facial rig and actor's facial proportions. We analyze similarities of motions spaces defined by blendshapes and expressive training sequences of a given actor to formulate the problem of transferring blendshapes of facial rigs to the actor as a special case of manifold alignment. The approach requires sparse correspondences between characters only, making this approach efficient for retargeting marker-less and marker-based motion capture as well as animation transfer. The system performs blendshapes face animation for calculating time-varying weights to reproduce facial expressions on target rig using delta-blendshapes formulation, where the neutral expression is subtracted from the blendshape expressions to yield a displacement field. The retargeting system mainly transfers an actor's performance capture to the target character, using mostly marker-based optical motion capture. The approach is not limited to motion-based, as any marker-less retargeting can be easily converted to motion-based performance capture by tracing a subset of marker vertices through time. The system improves the blendshape transfer from the facial rig to the actor's proportions by aligning their facial expression manifold. They evaluate their approach by comparing it to the RBF-proportion matching and example-based facial rigging. In all cases, this approach shows great results, and in almost all the expressions, the intensity is restored, whereas in RBF-proportion and example-based the quality degrades with a higher degree of stylization. This approach is suitable for cross-mapping as the approach won't respond well to a very cartoony behaviour of blendshape model. This approach also fails to address the transfer of fine details in the facial expressions.

Table 1: Summarization of papers in 2.1

Name of the system	Type of System	Issues in the system	Results/ Contributions
Performance Driven Facial Animation using Blendshape Interpolation [1]	Flexible 3D system that would create facial animation using a combination of motion captures data and blendshape interpolation.	- Assumption of linear interpolation is limited. - Better method could be found for the extraction of the key-shapes and then the key weights could exist.	- Motion capture process to drive the facial animation rather than animators animating by hand. - Resultant face model has the same facial motion as the original one, and the user has full control over the new face's appearance.
Vision-based Control of 3D facial Animation [2]	Real-time 3D facial tracking system that would be able to create a set of realistic facial expressions from a pre-processed motion capture database.	- Sometimes loses lip details. - Speech is not an input for the model. - Not enough user study to evaluate the efficiency and quality. - Expressions on the resultant model might not be as realistic as the original model because the synthesized motion is interpolated and adapted from the motion capture data.	- translate all these low-quality signals into high-quality 3D facial expressions. - developed a retargeting technique that would adapt to the synthesized motion in order to animate all the vertices of a model at run-time. - Automatically extracts 15 control parameters (mouth [6], nose [2], eye [2], and eyebrow [5]) that describe the facial expressions of the model.

Semantic 3d motion retargeting for facial animation [3]	3D realistic facial animation system that would have an ability to decompose facial motion capture data into semantically meaningful motion channels, using FACS.	None so far.	- Realistic 3D face model due to the combination of the high spatial resolution of 3D scanner and high temporal accuracy of motion capture data. - Automate the very time-consuming manual FACS annotation procedure. - Remove head motion using Iterative Closest Point (ICP) algorithm.
Feature points based facial animation retargeting [4]	3D system that would transfer facial animation in real-time.	- A lack of full-automatic detection of feature points on 3D virtual faces. - require improvement in the quality of our automatic rigging. - remove the need for marker in video tracker. - lacks the ability to capture and transfer fine details like wrinkles.	- proposed a procedural method called a quasi-automatic approach which computes for each vertex an influence weight to a feature point as per the distance. - developed a procedural technique that would be able to automatically rig the target face by generating vertices weights for the deformation process. - developed a testing application based on Pyramidal Lucas Kaneda Feature Tracker.

Characteristic facial retargeting [5]	Robust retargeting system that represents the characteristics of the target face model by using their hybrid method based on RBF and kCCA.	None so far.	- Computation efficient and robust. - The issue of the complexity of the cross-mapping is eliminated by the use of blended linear and non-linear retargeted models. - RBF understands and evaluates the training data efficiently, and kCCA outperforms for dense correspondence. - uses the Laplacian motion warping to enhance the characteristics of the retargeted face model.
Controllable High-fidelity Facial Performance Transfer [6]	Reconstructs the 3D retargeted face model animation in an optimized and efficient manner.	None so far.	- created an optimization algorithm which consists of face bases and weight vectors of the target sequence. - It will give users the ability to modify and control the retargeted face sequences in the spatial-temporal domain.
Facial Retargeting with Automatic Range of Motion Alignment	automatically retarget the facial expressions from actor to target character	- RBF-proportion and example-based the quality degrades with a higher degree of stylization.	- create an accurate parallel parameterization, in an unsupervised manner. - Efficient for retargeting marker-less and

[7]			marker-based motion capture as well as animation transfer. - Great results as compared to the RBF-proportion matching and example-based facial rigging.

2.2 Facial expression animation, weight alignment, and proportional editing

We know that the skin is supported by bones and multiple layers of muscle, which help with the movement of the face. However, we cannot completely simulate the neutrons, veins, muscles and bone of the face, but we can create a face model with a few dynamic parameters that can help emulate the basic facial movements. This is where facial animation can be used. The ideal goal of facial expression animation is to achieve realistic facial animation, in real-time, and to make the whole process automated and easy to reproduce/adapt for other users. Realistic facial animation is achieved by manipulating geometric (shape and deformations accounting expressions) and image (small-scale details that geometric deformation might miss like facial lines and hair) deformations. To get finer details from the face model, a few vertices can be set on the 3D face model either manually or automatically. Each vertex is assigned to a set of joint influences and their associated weights. You can bind any model to its skeleton using skinning, or you can model over a pre-existing skeleton to create its skin. When a model is bound to a skeleton using skinning, it then follows or reacts to the transformations of the skeleton's joints and bones. For example, if one bind a model's arm to its underlying skeleton using skinning, rotating the elbow joints causes the skin at the elbow to crease and pucker. By using skin weights, one can paint "weights" over a given model; this will provide weights that tell the vertices selected what percentage to be affected by the movement of the attached skeleton.

Proportional Editing is a way of transforming selected elements (such as vertices) while having that transformation affect other nearby elements. For example, having the movement of a single vertex causes the movement of unselected vertices within a given range. Unselected vertices that are closer to the selected vertex will move more than those farther from it (i.e. they will move proportionally relative to the location of the selected element). Since proportional editing affects the nearby geometry, it is very useful when one need to smoothly deform the surface of a dense mesh. By combining these two methods in a new innovative way, it might prove very useful to the task of facial expression transfer in order to create smooth and realistic facial animations.

Kalra et.al. [8] proposed an approach for deforming the facial skin surface using free form deformation. This system can specify and control facial expressions. We know that the skin is supported by the bone and multiple layers of muscle, which help with the movement of the face. In here we are not going to simulate the neutrons, veins, muscles, and bone of the face, but we would a model with a few dynamic parameters that can help emulate the basic facial movements. The Free Form deformation (FFD) technique is used to deform the solid geometric model in a free manner and to any degree. The deformation is applied to clear, flexible plastic and embedded in the object. The FFD starts to map out using trivariate tensor product Bernstein polynomial. This technique can achieve one more degree of freedom of manipulating deformations by changing the weights at the control points. For example, when all the weights are at control points in unity, they form a basic FFD, and when deformation is applied, it is possible to change the control points and weights at control points to produce a more realistic deformed model.

The landmarks can be obtained automatically or set on a 3D facial model manually, for the purposes of quick testing and proof of concept, the latter is attempted beforehand to confirm successful results of using these landmarks on the facial expression transfer system implemented. Lu and Jain [9] proposed a robust 3D system for facial surface modelling and matching of 2.5D facial scans in the presence of both non-rigid deformations and pose changes to a predefined 3D face model with neutral expressions. There are two models built: expression-specific and expression-generic. Out of these two, the expression-specific is better in deforming models but requires more computation. To model the expressions on the face, they derived three landmark layers. The first layer is a fiducial set of nine landmarks (two inner eye corners, two outside eye corners, two mouth corners, the nasion, nose tip, and sub nasal). The second layer was based on the first layer's landmarks where the geodesic distance and corresponding path between two fiducial landmarks is calculated using fast matching algorithm. The derived paths are used to make new extracted landmark points. Finally, the third layer is constructed over the new extracted landmarks. The system shows that three layers are enough for the modelling expressions. The resulting landmark set includes fiducial landmarks (9 points), first-layer landmarks (34 points), second-layer landmarks (40 points), along with the chin point (1 point), and mouth contour (10 points). The main issues faced by the system are fully automatic extraction of landmarks for deformation modelling (in the training stage), reducing the computational cost of matching, deformation learning using a larger control group, and evaluating the performance on the full FRGC v2.0 database. This system increases the 3D face matching accuracy by 7-10 points on FRGC v2.0 3D benchmark and the MSU Multiview 3D face database with expression variations.

Facial Action Coding System (FACS) is time-consuming, self-instructional and subjective which defines a visible facial movement of underlying muscles like the nose, eyebrow, eyelids, many more. It requires extensive training to encode ambiguous expressions. Hamm et.al. [31] developed an automated FACS system to overcome the issues. “The system would automatically track faces in a video, extracts geometric and texture features, and produces temporal profiles for each [31]”. The generated profiles would compute frequencies of one or more action units (AU). Action units are actions performed by each muscle. The system would be able to automatically deduce measures of flat and inappropriate facial effects. “The system observed that there are continuous temporal profiles of AU likelihood over the video period [31]”. The system compared the temporal profiles of each emotion to subjects who showed different characteristics. The system observed that subject healthier control had low flatness and inappropriate measures, whereas the rest of the subjects were higher in both. The system caused issues in the degree of flatness and inappropriateness of expressions across different emotions, and in the future, they would focus on finding a correlation between automated and observer-based measures.

Amini and Lisetti [10] developed a new open source software called HapFACS, and an API that would generate FACS-based facial expressions on 3D characters that have accompanying lip-synchronized animation abilities. The user can produce either videos or images repertories of realistic FACS-validated facial expressions. And the API helps the user to easily animate speaking characters with realistic real-time expressions into their own applications, without any professional experience. The main contribution of this system is that it gives control of 49 FACS action units with all intensity levels, able to activate the system unilaterally or bilaterally to animate faces with either single or multiple AUs, and compatible with any supported character in the 3D-system. The lack of difficult AU combinations, non-linear changing of intensities and lip-synchronization cause the performance of this system to drop. HapFACS is accessible, expressive, and easy to integrate with other applications. It is able to recognize correctly with an average of 90.48%.

Zollho¨fer et.al. [32] developed a robust non-rigid automatic algorithm that would generate a realistic 3D face model with textures. The input for the system is either a depth scan or an RGB image using the Kinect sensor. The algorithm is highly accessible and efficient as it doesn't require any user interactions. This feature caused issues as “ICP algorithm tends to converge into a local minimum if the deformation between the source and the target shape is too large [32]”. To estimate the alignment of the template mesh with a recorded point cloud, the system automatically detects four feature points. The resultant models can be easily analyzed, manipulated, or animated. The reconstruction of the face model took on average 18 seconds, with a maximum deviation from ground truth to be 4.7mm and average deviation to be 2mm. There is a lack of geometry and texture extraction information which hinders the expressibility of the model.

The definition of AU’s is an important concept since it allows direct control over the muscle movements that result in visible facial movement. When it comes to facial expression transfer, these concepts help with the transfer of specific facial segments to create realistic transformations from one expression pose to the next.

Cao et.al. [11] and Weise et.al. [33] focused on building a performance-based real-time facial animation model. Both models can handle occlusions and changing illumination conditions. Weise [33] stated that there are issues in complex acquisition and substantial manual post-processing like long turnaround time, the production cost for high-quality animation, and so on. To address these issues, they developed a performance-based system that would allow users to control expressions on a model in real-time. They used the depth maps and video from the Microsoft Kinect camera to perform the facial animation and make user-specific expression blendshapes, which was adapted by [11]. It is easy to deploy, and it can facilitate new applications. To avoid intrusiveness caused by Kinect due to invisible, infrared light while capturing facial data, the user performs an experiment in a non-intrusive environment with commercial 3D sensors. This device is very simple, but it causes high noise levels while acquiring data. Their main contribution was a novel face tracking algorithm that they developed to map low- quality 2D images and 3D depth maps in order to make the animation more realistic. It would combine 3D geometry and 2D textures registration with pre-recorded animation prior in a single optimization. The animations are added prior to temporal coherence in training, which is adapted by [11]. Without using any face markers, intrusive light, or hardware, their system could reconstruct a 3D facial dynamic in real-time. The Gaussian mixture model is trained using 9,500 pre-generated animation frames, which is adapted by [11], and depending on the size of temporal coherence, it might take 10-20 minutes to compute. It was found that few of the differences in expressions were not being recorded because of the limited amount of geometry and motion detail. There was mandatory user support required for registering lip and eye features. The system wasn't able to recover small scale or subtle movements like wrinkles, and there was no detection for eye, teeth, hair, or tongue. They are going to focus on the issues and implementing AAM and automated offline processing. The system is highly practical as it follows properties like usability, robustness, and performance, like [11], which has an addition of computational efficiency. Their system is relevant for their expression transfer into a digital avatar. In facial expression transfer, expression from a database is acquired and must be applied to the face model, the database model will not match with the custom model, and so a transfer method of this expression to a different model is needed.

[11] based his model on 3D shape regression and followed the system by [33]. The basic mechanism of the model was to infer 3D positions of facial landmarks using a regressor from the 2D video frame of a web camera, and then these points, the expressions are extracted by using a user-specified blendshape model. The 2D shape regressor automatically detects positions of facial landmarks, and when combined with 3D facial shape recovery algorithm, the system can estimate 3D facial shapes and use it to initiate a regression process. It takes up to 45-minutes to perform experiment first time and it will capture 60 images for each user. The main pipeline defined for the system is divided into three coordinate systems: object coordinate, camera coordinate and screen coordinate. Their main contribution was a 3D regression algorithm that would learn an accurate, user-specific face alignment model from a set of training data. By performing a sequence of predefined facial poses and expression, the algorithm would generate from images of a user. When tracking fails, the system can use the same process. An advantage of this model is that the run-time performance is high because it is independent of the video frame resolutions and it is defined by the number of initial shapes and landmarks. The issue in this model is that it is a two-step process when it should have been one-step to improve performance. The reasons behind it are that there are unexpected results as the direct regression doesn't guarantee expression coefficient, and they don't know how animation was before regression, which is essential for temporal coherence tracking. The landmark position is handled in the same manner in the two-step process, whereas in the

one-step process, rigid transformation and expression coefficient are dealt with in different ways as they are in different spaces. They are going to focus on building it in the one-step process.

Facial expression synthesis is a way of generating new faces using existing face while keeping a few of the distinct features. This work has mostly been done in 2D, and it is recently shifting towards 3D as it would hold more geometric shape data, and it would be invariant to pose and light changes. Agianpuye and Minoi [12] surveyed recent methods and approaches, outlining their applications and benefits. Interpolation can easily create fast facial expressions, but the resultant facial expressions are limited to the realistic facial configuration. Mass-spring Mesh creates a realistic facial expression by applying force to elastic meshes via muscle arcs. But the facial expression synthesis cannot be automated. Vector Muscle can model the muscle action of the skin, exploiting the delineated deformation field, but there is no guarantee for efficient placement of vector muscles [12]. Layered Spring Mesh can model detailed structures and dynamics of face to achieve high realistic facial expressions. While simulating volumetric deformations with 3D lattices, it requires extensive computation. Spline pseudo muscles can support even and flexible deformations to reduce complexity, but it causes deformation difficulty as it is required to be finer than patch resolution. Morphing is quick and efficient computation because it is integrated with GPU pipeline, but that makes it highly dependent on GPU, and if the GPU has low specification the processing speed will be reduced. MPEG-4 is automatic, fast, robust and accurate 3D face reconstruction model with not very realistic facial expression generation.

Table 2: Summarization of papers in 2.2

Name of the system	Type of System	Issues in the system	Results/ Contributions
Simulation of Facial Muscle Actions Based on Rational Free Form Deformations [8]	Deforming the facial skin surface using free form deformation.	None so far	- Specify and control facial expressions. - Able to one more degree of freedom of manipulating deformations by changing the weights at the control points.
Deforming Modeling for Robust 3D face matching [9]	Robust 3D system that transfers both non-rigid deformations and poses changes to a predefined 3D face model with neutral expressions.	- Fully automatic extraction of landmarks for deformation modelling (in the training stage), - reducing the computational cost of matching, - Deformation learning using a larger control group, and - Evaluating the performance on the full FRGC v2.0 database.	- Increases the 3D face matching accuracy by 7-10 points on FRGC v2.0 3D benchmark and the MSU Multiview 3D face database with expression variations. - Resultant face model has fiducial landmarks (9 points), first-layer landmarks (34 points), second-layer landmarks (40 points), along with the chin point (1 point), and mouth contour (10 points).

HapFACS [10]	- New open source software called HapFACS. - API that would generate FACS-based facial expressions on 3D characters.	- Lack of difficult AU combinations. - Non-linear changing of intensities. - Lip-synchronization causes the performance of this system to drop.	- Accessible, expressive, and easy to integrate with other applications. - It is able to recognize correctly with an average of 90.48%. - Gives control of 49 FACS action units with all intensity levels, able to activate the system unilaterally or bilaterally to animate faces.
3D Shape Regression for Real-time Facial Animation [11]	- 2D color image and a 3D depth map	Two-step instead of one-step: - Unexpected results as the direct regression don't guarantee expression coefficient, and they don't know how animation was before regression, which is essential for temporal coherence tracking. - Landmark position is handled in the same manner in the two-step process, whereas in the one-step process, rigid transformation and expression coefficient are dealt with in different ways as they are in different spaces.	- 3D regression algorithm that would learn an accurate, user-specific face alignment model from a set of training data - Run-time performance is high - Easy to use, robust, and computational efficiency.

3D Facial Expression Synthesis: A Survey [12]	- Survey of recent methods and approaches, outlining their applications and benefits. - 3D as it would hold more geometric shape data, and it would be invariant to pose and light changes.	- Interpolation produces facial expressions which are limited to the realistic facial configuration. - Mass-spring Mesh cannot automatically synthesis facial expression. - Vector Muscle provides no guarantee for efficient placement of vector muscles. - While simulating volumetric deformations with 3D lattices, Layered Spring Mesh requires extensive computation. - Spline pseudo muscle causes deformation difficulty as it is required to be finer than patch resolution. - Morphing is highly dependent on GPU, and if the GPU has low specification the processing speed will be reduced. - MPEG-4 has not very realistic facial expression generation.	- Interpolation can easily create fast facial expressions. - Mass-spring Mesh creates a realistic facial expression by applying force to elastic meshes via muscle arcs. - Vector Muscle can model the muscle action of the skin, exploiting the delineated deformation field. - Layered Spring Mesh can model detailed structures and dynamics of face to achieve high realistic facial expressions. - Spline pseudo muscles can support even and flexible deformations to reduce complexity. - Morphing is quick and efficient computation because it is integrated with the GPU pipeline. - MPEG-4 is automatic, fast, robust and accurate 3D face reconstruction model.

2.3 MPEG-4

MPEG-4 (Motion Picture Experts Group) is an object-based first international multimedia compression standard that allows for encoding of different Audio-Visual Objects (AVO) in the scene independently. It specifies a neutral face model with a few feature points as reference points, and a set of Face animation Parameter (FAPs) representing each facial action for deforming the face model. Deforming a face model will generate facial animation sequence, with specific FAP values for each time instant. The FAP value is depicted as the magnitude of respective action like deforming half eye vs. full eye open.

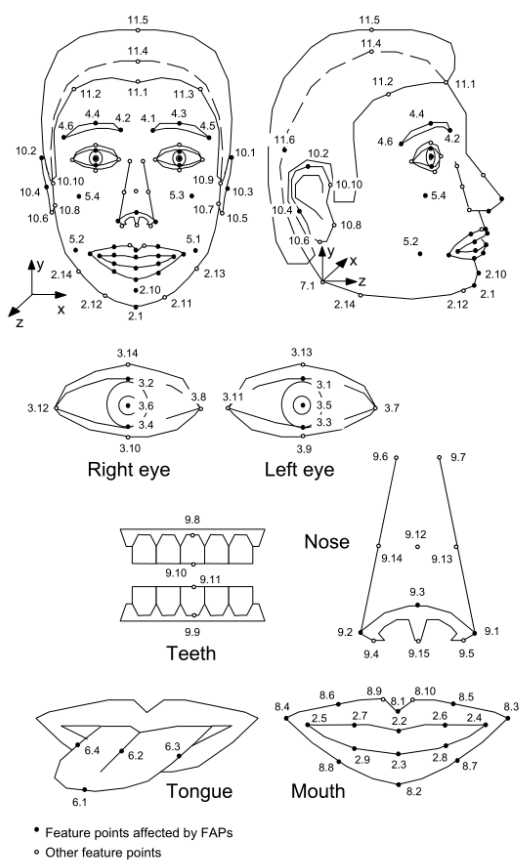


Figure 1: MPEG-4 FAPS [14]

Kshirsagar et.al. [13] developed a fast-geometric mesh deformation algorithm. This algorithm is robust and based on feature points, which will be used for deformation and animation on any generic surface mesh. To achieve smoothness and realism, each vertex should be able to control not only the neighbouring feature points but other feature points in the surrounding. The algorithm is divided into two parts. Initialization step, where the distance between the feature points and vertices are extracted, the weight for each vertex is calculated as per the nearest feature point. Each vertex can have more than one feature point approaching it. And the Deformation step, which takes place at real-time animation for each frame. The algorithm uses 3 markers on the head to capture rigid head movements. "Once we extract the global head movements, the motion trajectories of all the feature point markers are compensated for the global movements and the absolute local displacements and subsequently the MPEG-4 FAPs are calculated [13]". The algorithm obtains a frame rate of 29 frames per second with 1257 vertices. The algorithm works well with extracting the facial features from an optical tracking and retargeting them to the synthetic face.

Ostermann [14] describes how facial animation can be achieved using MPEG-4. MPEG-4 can animate face model for MPEG-4 users by the help of Facial Animation Parameters (FAPs). "FAPs can be compressed using either differential pulse code modulation (DPCM) or discrete cosine transform (DCT) [14]". The animated face model can be downloaded using MPEG-4 Binary Format for Scenes (BIFS). Using Timestamps, a media stream can be attached to the face. Initially, MPEG-4 will define a face model in a neutral state. Then the FAPs will be initialized with a fraction of distance between the key features like eye separation, mouth-nose separation, and many more. Ideally, MPEG-4 defines 84 feature points on the neutral face which serve as a spatial reference for FAPs. To animate a face over low bit rate communication channels, MPEG-4 uses temporal prediction to encode FAPs.

Hong et.al. [15] talk about real-time speech-driven face animation, and they developed an iFace system for facial animation and modelling. Using a set of labelled face deformation data, a visual representation called Motion Unit (MU) is learned. “A facial deformation can be evaluated by a linear combination of MUs weighted by the corresponding MU parameters (MUPs) [15]”. MU-based FA is compatible with MPEG-4 FA. The main issue when performing speech-driven face animation is AVM. The MPEG-4 FA standard defines 68 MPEG-4 FAPs, out of which two are high-level parameters and others are low-level parameters. MPEG-4 FAPs do not encode the information about the correlation between facial FP, and it does not specifically detail spatial information of face deformation. In the iFace system, a set of facial deformation is manually designed for the model. This system is an effective solution for very low bit-rate communication.

Tsapatsoulis et.al. [16] talk about a method that would be able to generate realistic emotions in HCI from primary and intermediate expressions, using MPEG-4 FDPs and FAPs. The method is a two-fold process: firstly, modelling primary expressions using FAPs, and secondly, defining a rule-based process for both the archetypal and intermediate expressions. Initially, the system will use FAPs to get the description of the archetypal expressions. Then the range of FAPs variations is estimated in the archetypal expressions. After the estimation, the intermediate expressions are modelled. Finally, the emotion model is created by understanding the emotions sequence. The two main issues that arise from the archetypal expressions are the estimation of FAPs involved and definition of FAP intensities. To verify the vocabulary of FAPs in archetypal emotions, the Ekman's and MediaLab's datasets are analyzed. The system is extensive with respect to the modelling of non-archetypal expressions. This system can serve as a visual part of the emotion recognition system or a client-side application for creating synthesis expressions.

Petajan [17] talks about MPEG-4 standards and FA for facial animations. They describe the standards as a collection of media coding tools that can be combined into profiles. Each profile defines distinct levels that can limit the decoding complexity of converting bit streams into objects. A decoder must be MPEG-4 compliant. "The MPEG-4 standard contains component codecs, object definitions, and interface specifications, and the FA standard contains a FAP codec, Face object, text-to-speech (TTS) interface [17]". The FAP definition and codec is specified visually because it describes the frame-based motion of the face. The Single Face Visual Profile specifies a minimum face for MPEG-4's output. The Simple Face Profile must show few parameters like FDP points must be in sync with FAP definition, the neutral face FDP points must be separated according to FBA conformance tests table, and there should be a minimum rendered frame rate as per the FBA conformance tests table. Level 1 is designing for wireless devices, and level 2 is for PCs, game consoles, and many more.

Zhang et.al. [18] discuss a probabilistic framework for the reproduction of facial expressions from a synthetic face model with MPEG-4 FAPs low bit rate communication. To unify facial expression analysis and synthesis into one, the system uses a coupled Bayesian network (BN). While analyzing, the system casts FAPs and

FACS into a dynamic BN to define the uncertainties in FAP extraction. The modelling of dynamic facial expressions also takes places in the analysis phase. “At the synthesizer phase, a static BN will reconstruct the FAPs and their intensities [18]”. The main contributions of the framework are “a very low bitrate in data transmutation, the facial expression is inferred by both spatial and temporal inference so that the quality of facial expressions is less affected by misdirected FAPs, and more realistic expressions are reproduced by modelling dynamic of human expressions [18]”. The process of MPEG-4 FAP defining 68 FAPs is like the paper by [15]. The issue faced by the system is the reproduction of individuality of facial expression. Because the semantic relations between the FAPs and the facial expressions are parameterized by person-independent muscular actions in the face.

Table 3: Summarization of papers in 2.3

Name of the system	Type of System	Issues in the system	Results/ Contributions
Feature point based mesh deformation applied to MPEG-4 facial animation [13]	Robust and fast geometric mesh deformation algorithm based on feature points.	None so far.	- To achieve smoothness and realism, each vertex should be able to control not only the neighbouring feature points but other feature points in the surrounding. - Obtains a frame rate of 29 frames per second with 1257 vertices. - Works well with extracting the facial features from an optical tracking and retargeting them to the synthetic face.
Face Animation in MPEG-4 [14]	How facial animation can be achieved using MPEG-4	None so far.	- MPEG-4 can animate face model for MPEG-4 users by the help of FAPs. - MPEG-4 defines 84 feature points on the neutral face which serve as a spatial reference for FAPs. - To animate a face over low bit rate communication channels, MPEG-4 uses temporal prediction to encode FAPs.

IFace [15]	- Real-time speech-driven face animation. - They developed an iFace system for face animation and modelling.	- When performing speech-driven face animation is AVM.	- MU-based FA is compatible with MPEG-4 FA. - A set of facial deformation is manually designed for the model. - This system is an effective solution for very low bit-rate communication.
Emotion Recognition and Synthesis Based on MPEG-4 FAPs [16]	- Method that would be able to generate realistic emotions in HCi from primary and intermediate expressions, using MPEG-4 FDPs and FAPs. - It uses Ekman's and Media Lab's datasets.	- Estimation of FAPs involved. - Definition of FAP intensities.	- Two-fold process: firstly modelling primary expressions using FAPs, and secondly, defining a rule-based process for both the archetypal and intermediate expressions - Serve as a visual part of the emotion recognition system or a client-side application for creating synthesis expressions.
MPEG-4 Face Animation Conformance [17]	MPEG-4 standards and FA for facial animations.	None so far.	- Single Face Visual Profile specifies a minimum face for MPEG-4's output. - Simple Face Profile must show few parameters like FDP points must be in sync with FAP definition, the neutral face FDP points must be separated according to FBA conformance tests table, and there should be a minimum rendered frame rate as per the FBA

			conformance tests table.
Dynamic Facial Expression Analysis and Synthesis With MPEG-4 Facial Animation Parameters [18]	Probabilistic framework for the reproduction of facial expressions from synthetic face model with MPEG-4 FAPs low bit rate communication.	Reproduction of individuality of facial expression	- To unify facial expression analysis and synthesis into one, the system uses a coupled Bayesian network (BN). - a very low bitrate in data transmutation, the facial expression is inferred by both spatial and temporal inference so that the quality of facial expressions is less affected by misdirected FAPs, and more realistic expressions are reproduced by modelling dynamic of human expressions. - The defining process of MPEG-4 FA defining 68 FAPs is similar with the paper by Hong et al. (2002).

2.4 Transfer of expressions

Synthesis and retargeting of facial expressions require manual work to achieve realistic expressions due to difficulties faced in capturing high-quality dynamic expression data. Wang et.al. [19] developed a data-driven approach that comprises of four parts: "high speed and accuracy capture of moving faces without the use of markers, very precise tracking of facial motion using a multi-resolution deformable mesh, a unified low dimensional mapping of dynamic facial motion that can separate expression style, and synthesis of novel expressions as a combination of expression styles [19]". The relation must be established between data in different frames of both the same and different faces, to use the produced data for motion analysis and retargeting. The system is accurate and has the resolution to capture and track details about expression. The system can transfer the motion style of one person to another, irrespective of their face geometry. By combining 3D geometry and motion mapping, the system can synthesize new face motions. However, the system lacks skin reflectance modelling under light variations and specialized interior mouth and lip model for a large open mouth.

Vlasic et.al. [20] state that "a multilinear model can be calculated from a Cartesian product with statistical analysis, but with careful pre-processing of the geometric data set to secure one-to-one correspondence and to minimize cross-coupling artefacts". They developed a system called Face transfer that would be able to map a video recording of one actor and transfer it to another because it is based on a multimodel of 3D mesh. The methods extract visemes, expressions, and 3D pose from the video, which is then turned into a detailed 3D textured mesh for a target. These parameters can be changed easily as they are not dependent on each other to operate. The movement of target facial expressions is adjusted by the underlying face model. The model provides a good source of synthetic actors in the video, and it is very easy to edit and rewrite applications.

Yang et.al. [21] developed a robust neighbour-expression transfer (NET) model which is a hierarchical method to transfer expressions to sketches using motion retargeting. Their main objective is to model spatial relations among sparse facial features [21]. The system is able to reconstruct facial expressions from noisy sources by observing the behaviour of neighbouring expressions. The motion vectors are adjusted as required while transferring from source to target face. According to statistical results on manually labelled truth data about 10% SDR is an acceptable result which is close to reality, and by the ASM 20% SDR is achieved, which affect visual effect [21]. This system is considered best against any noisy source, and it can produce vivid expressions from retargeting motion vectors from the source. The only issue caused by the system is that it only covers the frontal face. They are focusing on making the transfer of expression for 3D sketches in the future.

Minoi et.al. [22] developed a robust system that would transfer realistic expressions to a 3D face model captured by frontal photographs with a neutral face. The databases used by the system are the Notre Dame 3D face database, the Imperial College VisionRT 3D face database, and the FERET face database. The system is using an analysis-by-synthesis approach based on a statistical model to encode facial information from recovered 3D face surface. To implement new expressions, the system uses a statistical discriminant model that would divide the training data into two classes and then determine a relationship between them. The 3D face model created can render in a variety of poses and illumination. “The SDM approach extracts expression discriminant information efficiently, providing a gradual transformation on the 3D faces [22]”. The main advantage of the system is that the variety of expressions can be easily generated without performing intense changes in the 3D database. The size and range of the selected databases for the system cause an efficiency issue in the system.

Huang and De La Torre [23] developed a two-step FAT system that would be able to transfer expressions from a video source of a person to another target person. To have a successful method, there should be training set with different expressions from different subjects, but labelling expressions can be very time to consume, not realistic, and it can cause errors. The main objective of the system is to transfer expression to a different subject, and not to reconstruct appearance from the shape of the subject. The system works very nicely with Cohn-Kanade and the RU-FACS databases. The main issues faced by this system were the direct transfer of the face shape and appearance from the source to the destination results in rendered output, transfer of textures remains is unsolved due to high dimensionality and potential nonlinearity, factorization is difficult, and uncertainty of spontaneousness of facial expressions and poses. “The learning method does not require the correspondence of expressions across training subjects [23]”.

The larger the set of blend shape expressions, the better the quality of the animation, but creating enough blendshapes takes up a ton of time and effort. Pawaskar et.al. [24] address this issue by developing a complete automated template system that would build a 3D face model comprising of a newly generated set of blendshapes with various expressions on a neutral expression face mesh. The system transfers the face animation generated for the template character to a new face model. Also, the system allows the blending of the template face mesh to a new face mesh model while animating expressions, using the triangle-wise correspondence. The system will begin by registering the template mesh and the face scan, and then they will deform it into another shape while keeping its geometrical features and topology the same. The system uses a non-rigid ICP (iterative closest point) algorithm to register the produced template mesh model onto the target face mesh. The system will use pre-vertex affine transformation to make the template mesh as close to the target mesh as possible, and a Laplacian regularization is used to minimize the distance between the transformation matrices of the adjacent vertices. The template is implemented into Maya (3D authoring tool) as a plug-in. The system provides good results for transformation deformation based on identity without adding virtual triangles, but it cannot handle a very wide variation from the template.

Orvalho [25] developed a system that would automatically transfer facial rigs and animation from one face model to another, independent of their shape and appearance. The process of transfer is 90-99% faster than the traditional methods, thus creating high-quality facial animation. "The system converts 3D models into puppets that experienced animators can control using a generic facial rig definition and new deformation method [25]". This system also supports easy reusability of existing scripts. While the system performs well, there are few obstacles. The system verifies if the source and target model have identical rigs which would allow the system to perform direct mapping of attributes between the two, but if that's not the case, the results will not be as expected. Secondly, for extreme cases the source and target model will have to be symmetric, and the eyes and mouth areas needs to be slightly open for expected results.

Zhang and Wei [26] developed a novel approach for synthesizing and transferring the dynamic expression from source to target model. This approach will maintain the facial appearance of target model as well as expression deformation from the source model. The source model's expression is captured by TensorFace which is trained on a small sample scale. To achieve smooth shape variations, the system interpolates the aligned sequential shapes of different expressions by wrapping the neutral expression into other expressions. The facial details are obtained and transferred using nonlinear TensorFace to the wrapped dynamic expression faces. The experiment is evaluated by using Cohn Kanade facial expression database and it depicted that the proposed system has better results than state-of-the-art methods, but the expressions transferred lack realistic factor in few of the expressions like mean, anger, so on.

Gunanto et.al. [27] developed an automatic pre-processing transfer system for facial animation using Feature Point Cluster. The system uses a clustering process to perform clustering of objects on learning from the data of vertices that are located close to the point feature, it is known as k-Nearest Neighbour (k-NN) with some modifications in the definition of the value of k. To improve the production speed, the system is using Radial Basis Function (RBF) which is used for estimation and surface interpolation process. The system focuses on improvement of facial expression animations using feature point clusters and making the process easier and faster without the manual adjustment process. To deform the mesh in reference with the local transformation of the skeleton, the Linear Blend Skinning (LBS) is adopted. The complexity of the system can be improved by creating a real-time animated facial expression transfer system.

Bickel et.al. [28] developed a hybrid real-time animation system for highly detailed facial expressions based on decomposition of facial geometry into large- and fine-scale details. The system handles large-scale deformations with fast linear shell model through a sparse set of motion-capture handles or user-defined markers, and for the fine-scale deformation, a novel pose-space deformation technique is used which adapts the skin strain's sparse measurements to wrinkle formation using small example sets. However, there are some issues in this system like the animation is driven by handles or markers so the face model cannot react to forces, only to position constraints. Another issue is that the system has self-intersections.

Bouaziz et.al. [29] developed a real-time algorithm for face tracking on RGB-D sensing devices. This method doesn't require any user-specific training, calibration, or manual assistance. They also developed an optimizing algorithm which solves the dynamic tracking parameters for the detailed 3D expression model by the user. Using subspace parameterization of dynamic facial expression space, the system facilitates the real-time performance and robust computation. However, the system has issues like it is limited by the noise levels and resolution of the input device, and the geometric detail of blend shape template model defines the tracking accuracy.

Orvalho [30] developed a facial deformation system that contains main attributes from a generic rig, transfer them to different 3D face model and generate a facial rig based on an anatomical structure. This system is implemented in C++ as a plug-in for Maya. It gives the user an ability to setup any character, manipulate the model to adjust animation parameters, and animate the target models using generic rig's pre-defined animations. The system is defined by using a set of landmarks that handle specific facial features in order to deform them anthropometrically. The system depicts how a combination of labels and other deformation methods can adapt to muscles and skeletons from a generic rig to different face model. Initially, the method deforms the generic rig surface to match the topology of the target face model, then it adapts the muscles, skeleton, and attributes from the generic rig onto the target model, and finally it combines the transferred elements to the model to achieve an anatomic structure for physically-based animation. To deform the rig, the method uses a combination of linear and non-linear global transformation, along with local deformation to give more degree of freedom. This system speeds up the character setup and animation pipeline, and with the use of dense correspondence, the system uses only landmarks on surface as compared to other methods which use landmarks on skin, muscles, and skull. The system uses 10 landmarks for transfer for the human model and 12 for extreme cases like a goat. The automation of character setup process and support proposed plug-in for NURBS surfaces is some of the future works for the method.

Table 4: Summarization of papers in 2.4

Name of the system	**Type of System**	**Issues in the system**	**Results/ Contributions**
High-Resolution Acquisition, Learning, and Transfer of Dynamic 3D Facial Expressions [19]	A data-driven approach which must have a relationship established between data in different frames of both the same and different faces, to use the produced data for motion analysis and retargeting.	Lacks skin reflectance modelling under light variations and specialized interior mouth and lip model for a large open mouth.	- Transfer motion style of one person to another, irrespective of their face geometry. - By combining 3D geometry and motion mapping, the system is able to synthesize new face motions.
Face transfer [20]	Map video recording of one actor and transfer it to another because it is based on a multimodel of 3D mesh.	No issues so far.	- Parameters can be changed easily as they are not dependent on each other to operate. - Provides a good source of synthetic actors in the video. - Very easy to edit and rewrite applications.
Neighbor expression transfer (NET) [21]	- Robust, hierarchical method to transfer expressions to sketches using the motion retargeting. - Models spatial relations among sparse facial features.	- Only cover the frontal face. - Making the transfer of expression for 3D sketches in the future.	- Consider best against any noisy source. - Can produce vivid expressions from retargeting motion vectors from the source.

	- Reconstruct facial expressions from noisy sources by observing the behaviour of neighbouring expressions.		
Synthesizing Realistic Expressions in 3D Face Data Sets [22]	- Robust 3D system to transfer expressions with a neutral face. - Databases used: Notre Dame, FERET, and Imperial College VisionRT.	- Size and range of the selected databases for the system cause an issue inefficiency of the system.	- Able to render in a variety of poses and illumination. - Variety of expressions can be easily generated without performing intense changes in the 3D database.
Facial Action Transfer (FAT) [23]	- Transfers expression from one to another different subject. - Does not reconstruct. - Databases used: Cohn-Kanade and RU-FACS.	- Direct transfer of the face shape and appearance from the source to the destination results in the rendered output. - Transfer of textures remains unsolved due to high dimensionality and potential nonlinearity. - Factorization is difficult. - Uncertainty of spontaneousness of facial expressions and poses.	- System works nicely with both databases. - Learning method does not require the correspondence of expressions across training subjects.

Expression Transfer [24]	Complete automated template system that would build a 3D face model comprising of the newly generated set of blendshapes with various expressions on a neutral expression face mesh.	- Cannot handle a very wide variation from the template.	- Blending of the template face mesh to a new face mesh model while animating expressions, using the triangle-wise correspondence. - Template is implemented into Maya (3D authoring tool) as a plug-in. - Good results for transformation deformation based on identity without adding virtual triangles.
Reusable Facial Rigging and Animation: Create Once, Use Many [25]	Automatically transfer facial rigs and animation, independent of shape and appearance.	- System verifies if the source and target model have identical rigs which would allow the system to perform direct mapping of attributes between the two, but if that's not the case, the results will not be as expected. - Secondly, for extreme cases the source and target model will have to be symmetric, and the eyes and mouth areas needs to be slightly open for expected results.	- Process of the transfer is 90-99% faster than the traditional methods. - Easy reusability of existing scripts.

A realistic dynamic facial expression transfer method [26]	- Novel approach for synthesizing and transferring the dynamic expression from source to target model. - Captured by TensorFace.	- The expressions transferred lack a realistic factor in few of the expressions like mean, anger, so on.	- Evaluated by using Cohn Kanade facial expression database and it depicted that the proposed system has better results than state-of-the-art methods.
Facial Animation of Life-Like Avatar based on Feature Point Cluster [27]	Automatic pre-processing transfer system for facial animation using Feature Point Cluster.	- Complexity of the system can be improved by creating a real-time animated facial expression transfer system.	- Clustering process is performed by k-NN. - Production speed is improved by Radial Basis Function. - The deformation of the mesh is performed by Linear Blend Skinning.
Pose-Space Animation and Transfer of Facial Details [28]	Hybrid real-time animation system for highly-detailed facial expressions based on decomposition of facial geometry into large- and fine-scale details.	- The animation is driven by handles or markers so the face model cannot react to forces, only to position constraints. - The system has self-intersections.	- Handles large-scale deformations with fast linear shell model through a sparse set of motion-capture handles or user-defined markers. - Handles fine-scale deformation with a novel pose-space deformation technique is used which adapts the skin strain’s sparse measurements to wrinkle formation using

			small example sets
Online Modeling for Real-time Facial Animation [29]	Real-time algorithm for face tracking on RGB-D sensing devices.	- Limited by the noise levels and resolution of the input device. - The geometric detail of blendshape template model defines the tracking accuracy.	- doesn't require any user-specific training, calibration, or manual assistance. - Optimizing algorithm which solves the dynamic tracking parameters for the detailed 3D expression model by the user - Real-time performance and robust computation.
Transferring Facial Expressions to Different Face Models [30]	Facial deformation system that contains main attributes from a generic rig, transfer them to different 3D face model and generate a facial rig based on an anatomical structure.	None so far.	- Speeds up the character setup and animation pipeline. - Uses only landmarks on surface as compared to other methods because of dense correspondence.

In this work, the automatic transfer of facial expressions is further explored and a new technique to implement a smooth and aesthetically pleasing method is introduced. Some aspects of previous research in this area will be built upon, as well as a professionally modelled and animated 3D face model will be used as the source of predefined expression available. The aim is to provide an easy and ready to use tool within Maya 3D modelling software to create facial animations based on the transition of user-selected facial expressions.

Chapter 3: System Description

In this chapter, we will be elaborating on the proposed facial transfer system's designs, processes, and user-friendly capabilities.

We first start with an overview of the system as a whole; an initial proposed Weighted Cluster method for simple facial animation transfer, and finally an explanation of the new and improved Vertex-Based Animation Transfer methodology. This gives the reader a general understanding of the system as a whole. The chapter moves on with a description of the source model used, as well as the original proposed Weighted Cluster method for transfer of animation. An important aspect is then described, which is the landmark definition, where we introduce the method of selecting vertices to create reference points, using Vertex Matching and Landmark-Based Surface Subdivision. And finally, the transfer method, where the Geometric Transformations and creation of animations are applied to a target model using the transform data extracted from a source model. This portion includes transfer for individual pose-to-pose expressions and transfer of multiple expressions in an animation sequence.

3.1 System Overview

The system as a whole can be divided into three distinct processes: animation extraction, landmark definition with vertex matching, and facial animation transfer. Provided in figure 2 is a high-level architecture of the system.

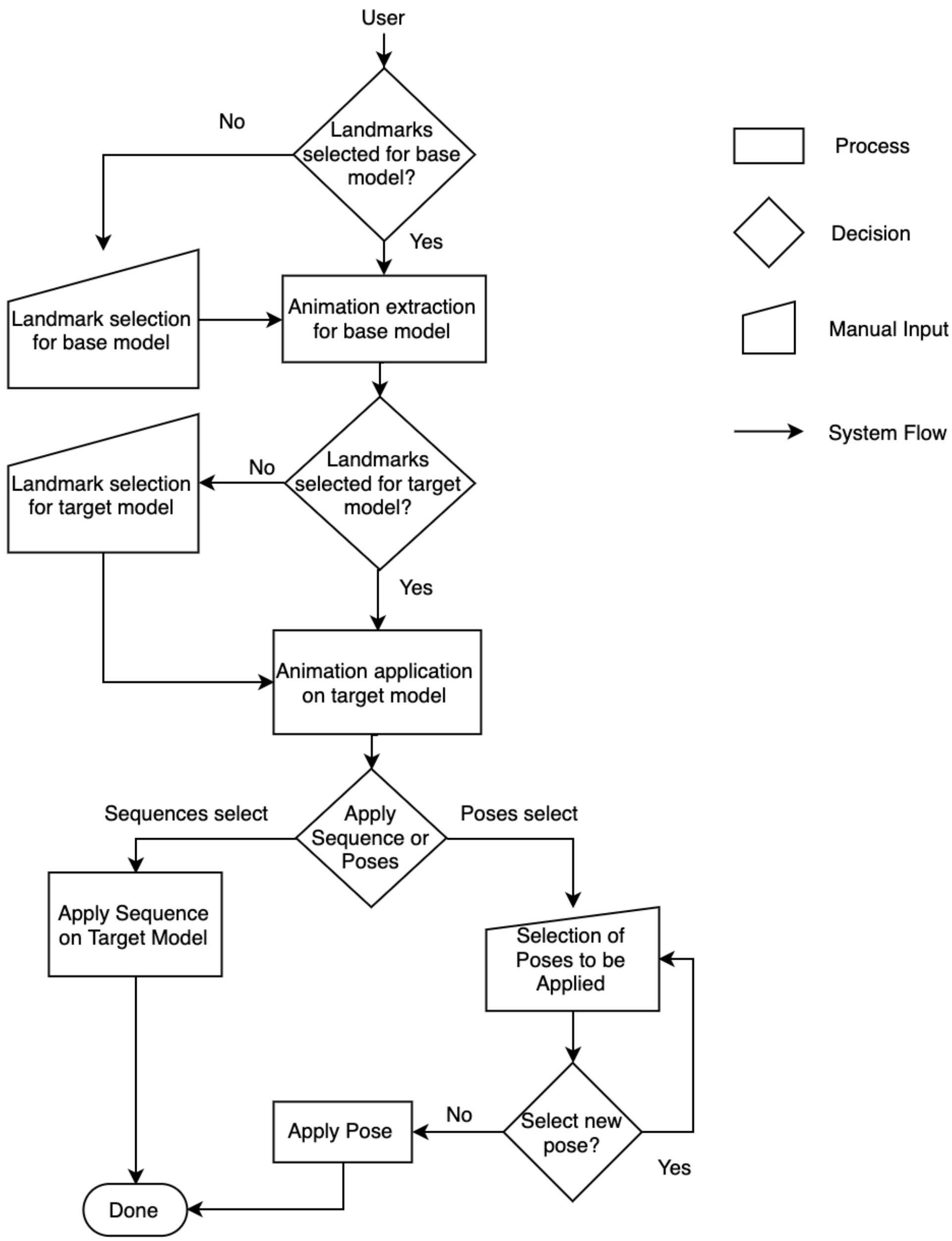


Figure 2 High-Level Architecture of the System

In figure 2, we have depicted a high-level architecture of the system. To start the process, the user would select landmarks on the base model, if not already selected. Then the system would perform the extraction process. Then the user will move onto selecting landmarks on the target model, if not already selected. Then the system will perform the application process where the vertices on both models will be matched to each other, this process will be explained in details in further chapters. Then the user will have to choose between application of poses or sequences. If they select poses, the user can apply as many poses, they would like till they are satisfied with the resultant animation. On the other hand, if they select sequences, they would choose 1 sequence from 4 sequences to apply. The extraction and application process will be explained in respective chapters.

Our system provides many features:

- **Generic**: The 3D face models used in the system or uploaded by the user can have any topology, vertex count, and/or vertex order. The face models can have a very low or very high number of vertices, as this system works with extreme cases as well.
- **Flexible:** The system can be easily modified to include the face models with different types of deformation methods. The current proposed system uses a source 3D face model with professional blendshapes, but it can easily be replaced with any other method like rigs and custom animations.
- **Independent:** The face model can have any shape or appearance, as long as it is human-like.
- **Reusable:** The source 3D face model is used for n-number of target 3D face models, as well as an individual target 3D face model can use different source 3D face model for facial expression animation transfer procedure.

3.2 The Source Model

The 3D facial model used as a source for the system is a professionally modelled facial mesh with multiple blendshapes to control all face muscles downloaded from Turbosquid. This gives the target user a set of professionally modelled facial expression to apply on any face model to create a single, multiple, or multiple overlapping animations. Furthermore, the model was handed to a professional animator to create four sequences of animations with multiple expressions in each sequence. The sequences were created by manipulating the weights of each blendshape applied to the model and recording it as keyframe animation. Below is a figure of the source 3D model.

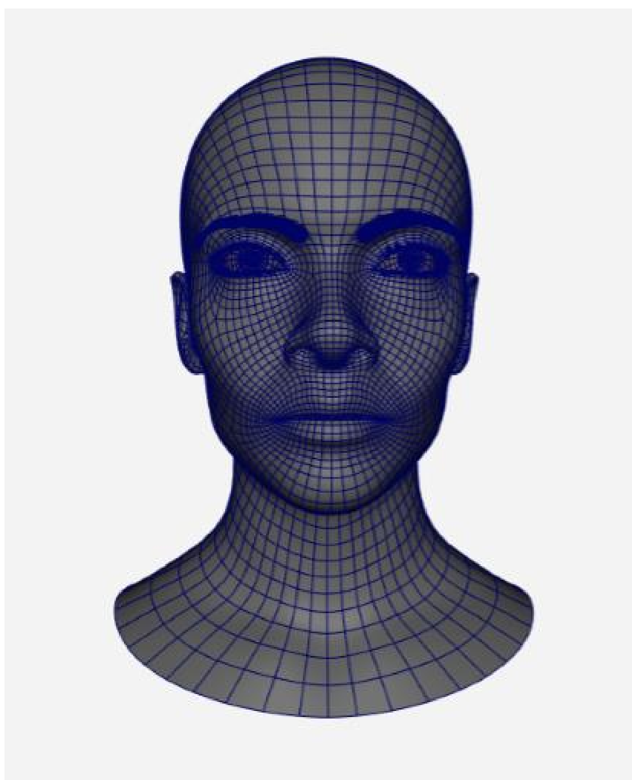


Figure 3: Source 3D Facial Model

As of now, the system was designed to work with base models that make use of any type of animation method to create animations. This includes blendshapes, animations done by face rigs, manual keyframe animation, or any other method. For the implementation in this paper, blendshape animation was the focus, as well as manual vertex manipulation with keyframe animations.

3.2.1 Transformation-based Animation Extraction

The extraction process begins with a scaling component. Due the difference in size of any given facial model, it is difficult to come up with a standard method of representing transformations that can be applied to any model. This is where scaling of the model comes in to keep a standard metric of transformations. We use the Euclid distance to calculate how large a given model is, and scale it so that it is approximately fits within the size of (1, 1, 1) in Maya units (x, y, z). This step is necessary since the scale transform of a model might indicate an identity vector scale, but the modelling process might have deformed the mesh's vertices and made it bigger or smaller without altering the scale. To fix this issue, we measure the distance from the chin to approximately eyebrow height of the model, and scale the model by a factor of 1.414214, which corresponds to the longest distance between any two vertices on a unit cube. In order to make this scaling work, the Maya scale transform must be (1, 1, 1) before the scaling is applied. After this scaling, it is ensured that the model will approximately fit into the space of (1, 1, 1). This will make both models relative to each other and that in turn will make the vertices of both models relative to each other.

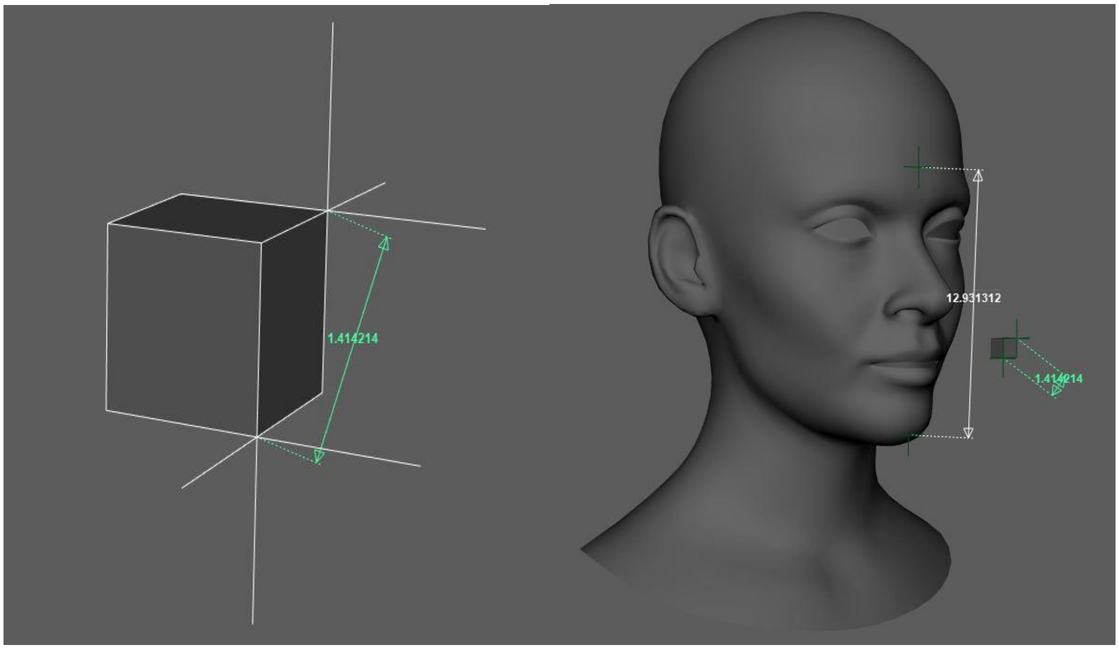


Figure 4: Left - Longest distance of the unit cube. Right - Size comparison of (1, 1, 1) facial model with unit cube.

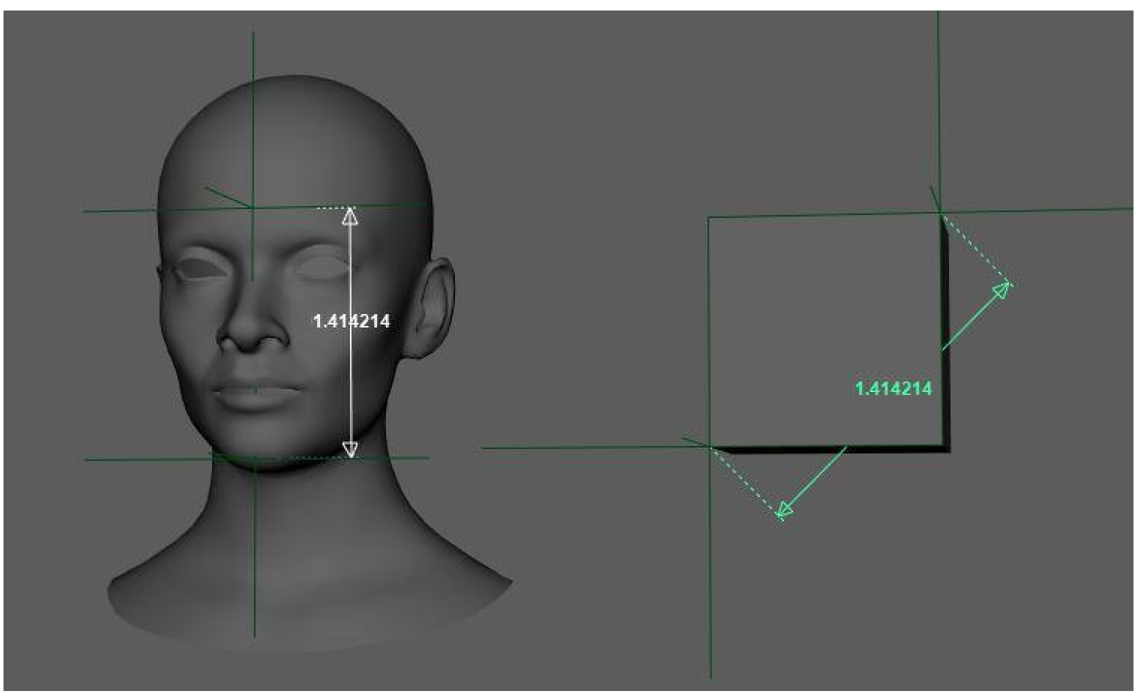


Figure 5: Comparison of facial model and unit cube after rescaling

Taking the example from the figure above, we see that the top right image shows a distance between the top and bottom vertices of 12.9313, this is with the scale transform of (1, 1, 1). By dividing the factor mentioned earlier by the distance, we get the new scale transform in order for the mesh to approximately fit in the (1, 1, 1) space.

For the extraction process, there is one last manual step, which is selecting the vertices that correspond to the landmarks. This is a guided component where a user is presented with a figure detailing which vertices should be chosen. These vertices, and later on landmarks, are based on MPEG-4 Facial Animation Parameters (FAPs). MPEG-4 is an object-based multimedia conversion standard that specifies a neutral face model with a number of feature points as reference points, and a set of FAPs representing each facial action for deforming the face model. MPEG-4 defines 68 FAPs, out of which two are high-level and rest are low-level parameters and 84 feature points on a neutral face which serve as a spatial reference for FAPs.

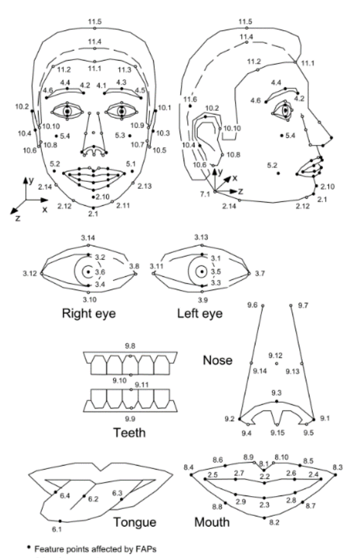


Figure 6: Graphical representation of MPEG-4 Facial Animation Parameters [14]

The proposed system has used a simplified version of MPEG-4 FAPs as a reference to select the vertices and create landmarks in the extraction and transfer process. There are 14 landmarks used across the face targeting the points needed for animation transfer in our system, these points are selected as they provide the best mesh subdivision which is explained later on. The landmarks are listed as follows.

Table 5: Landmarks (set of vertices) selected for the base and target face models

Landmarks	Description
1	Outer Right Eyebrow
2	Inner Right Eyebrow
3	Outer Left Eyebrow
4	Inner Left Eyebrow
5	Upper Right Eyelid
6	Lower Right Eyelid
7	Upper Left Eyelid
8	Lower Left Eyelid
9	Right Cheek
10	Left Cheek
11	Nose Tip
12	Upper Lip
13	Lower Lip
14	Chin

Given these reference points, the guided image is as follows:

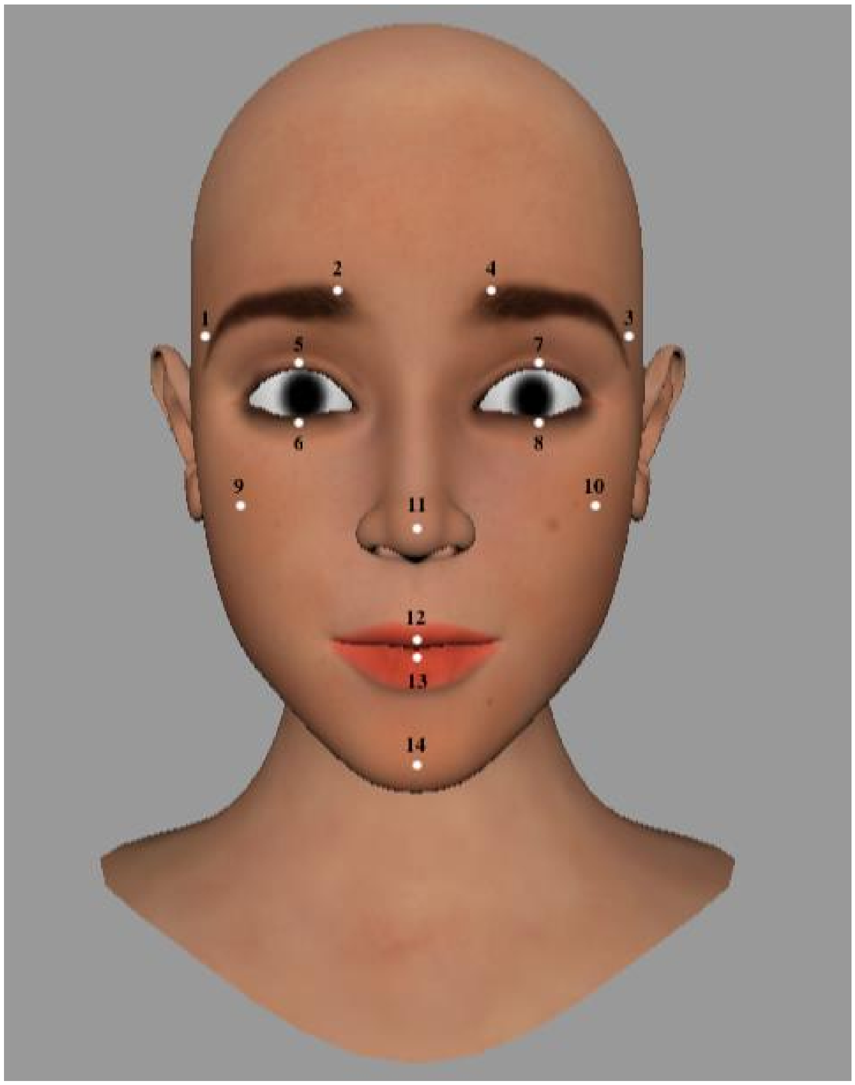


Figure 7: Vertex selection based on simplified FAPs

The system is flexible to allow different facial feature landmarks to be selected, as long as consistency is maintained between the landmarks selected on the source model and landmarks selected on the target model. The landmarks described in Table 5 must be selected in the order shown in Figure 8 and need not be exact, but an approximation of the area. The order is simply to give the system an order in which to compare between the base and target models. Due to Mesh Subdivision and a Vertex Matching algorithm, it is not crucial to select any vertex specifically; an approximation works for the purpose of expression transfer.

Once the vertices have been selected, the system continues with the extraction process. Extraction is divided into three different processes: poses and animation sequence.

3.2.1.1 Poses Extraction

Since the base model's animations used in this implementation are based on blendshapes, the algorithm finds all the target shapes in the blendshape applied to the base model. Each of these target shapes are the individual expressions (smile, blink, frown...). Target shapes are applied to the model by providing a weight, if all weights are set to 0, then no transformations have been made, and it is assumed the face model is in the neutral position. Neutral position does not mean that the global positions of the vertices are 0, so the system records the global position of all vertices in the model for the transformation calculation. Once the global position of vertices for the neutral face are saved, the system loops through all target shapes. In each iteration, the weight of the current target shape is set to 1, meaning, the expression is fully applied. At this point, we repeat the process of saving the global position of all the vertices, with the difference been that the vertex positions now represent the current expression instead of the neutral face. Now we have the necessary data for application of the individual poses.

3.2.1.2 Animation Sequence Extraction

In animation sequence, we have an animated base model. The animation consists of multiple keyframes where each keyframe consists of some set of transformations which modifies the global position of vertices. Similar to pose extraction, we must first save the set of vertices corresponding to the neutral face; it is assumed that the first frame contains the neutral face. The system then finds the next keyframe in the timeline and saves the positions of the vertices for that keyframe as well as a timestamp of the frame. This is repeated until the last keyframe is reached.

Once the extraction is completed, the model is returned to its original size by setting the scale to the original. Below is a graphical representation for the extraction process:

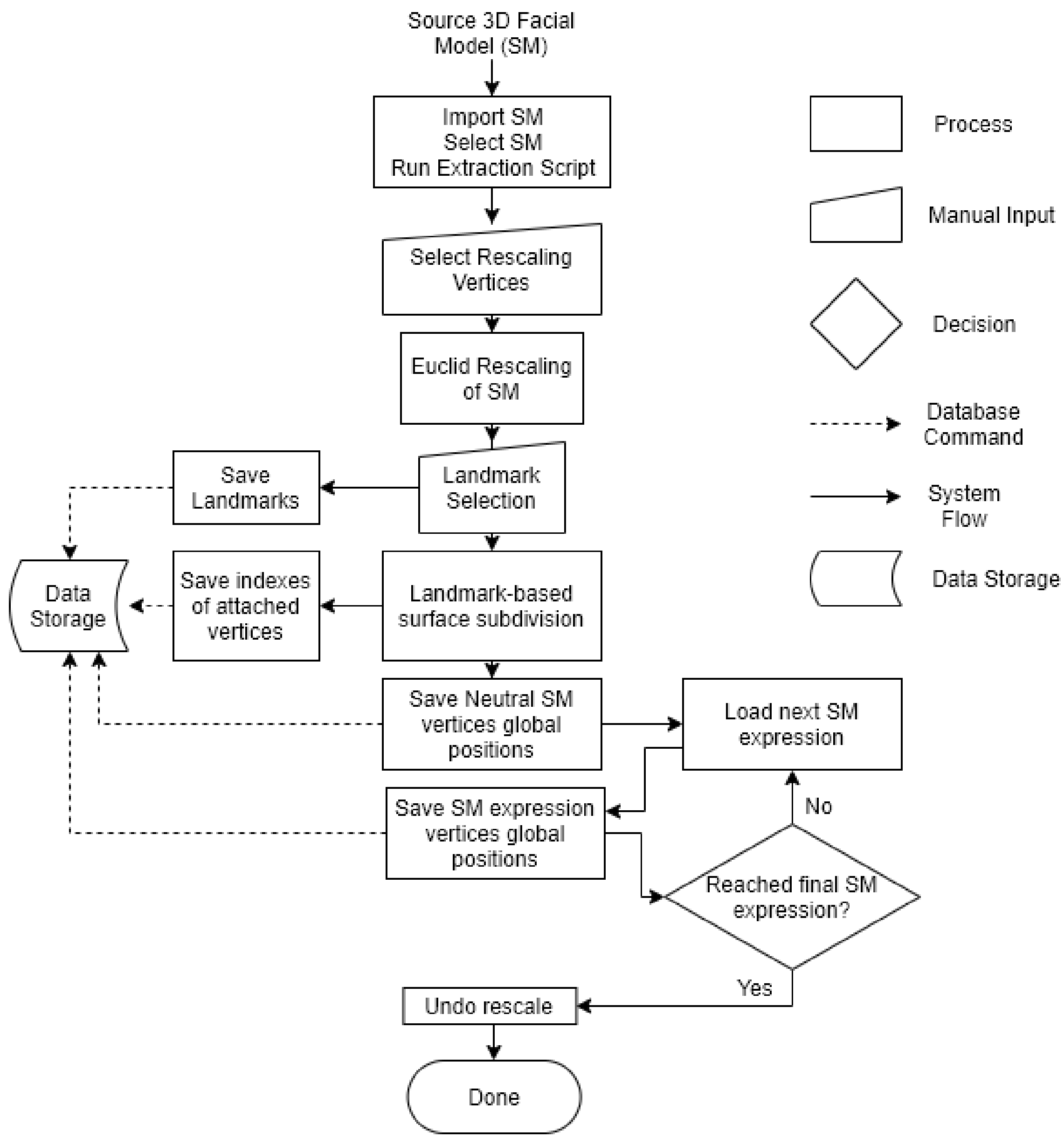


Figure 8 Pipeline describing the manipulation of source 3D facial model and animation extraction

In figure 8, we depict the detailed extraction process for the system. Initially, the user would start with a blank space and import the source model in Maya. Then they would select the source model and run the system script. The user will be prompted to select the rescaling vertices which are the two vertices on forehead and chin. Then using Euclid rescaling method, the source model will be rescaled into (1, 1, 1) scale. Then the user will be prompted to select the landmarks on the model as per the guide image provided with the step. The position of the landmarks does not have to be exact, just in the same area to maintain consistency between both the models. Then the system would save the landmarks along with the original scale of the model. After selection of landmarks, the surface of the mesh would be divided using the shortest path between the landmarks and all the vertices, that means that for each vertex, the system would check all the shortest paths from that vertex to all the landmarks, and whichever landmark has the shortest path, the vertex will be attached to that landmark. This process be performed for the vertices as each vertex has to be attached to a landmark. Then the system would save the indexes of all the attached vertices.

As the source model is based on blendshapes, there are targetshapes associated to it. Targetshapes correspond to movement in muscles like smile, blink, frown, etc., and its transformations can be manipulated by adjusting the weights between 0 and 1, where 0 corresponds to no transformation and 1 to full transformation. The system would set the weights to 0 so that it can save the global position of all the vertices in a neutral face of the model. Then it would set the weight to 1 for each targetshape to save the global position of all the vertices in the particular targetshape till the last one is saved. After that, the model is scaled back to its original scale and the user can proceed to application process.

3.3 Weighted Cluster Method

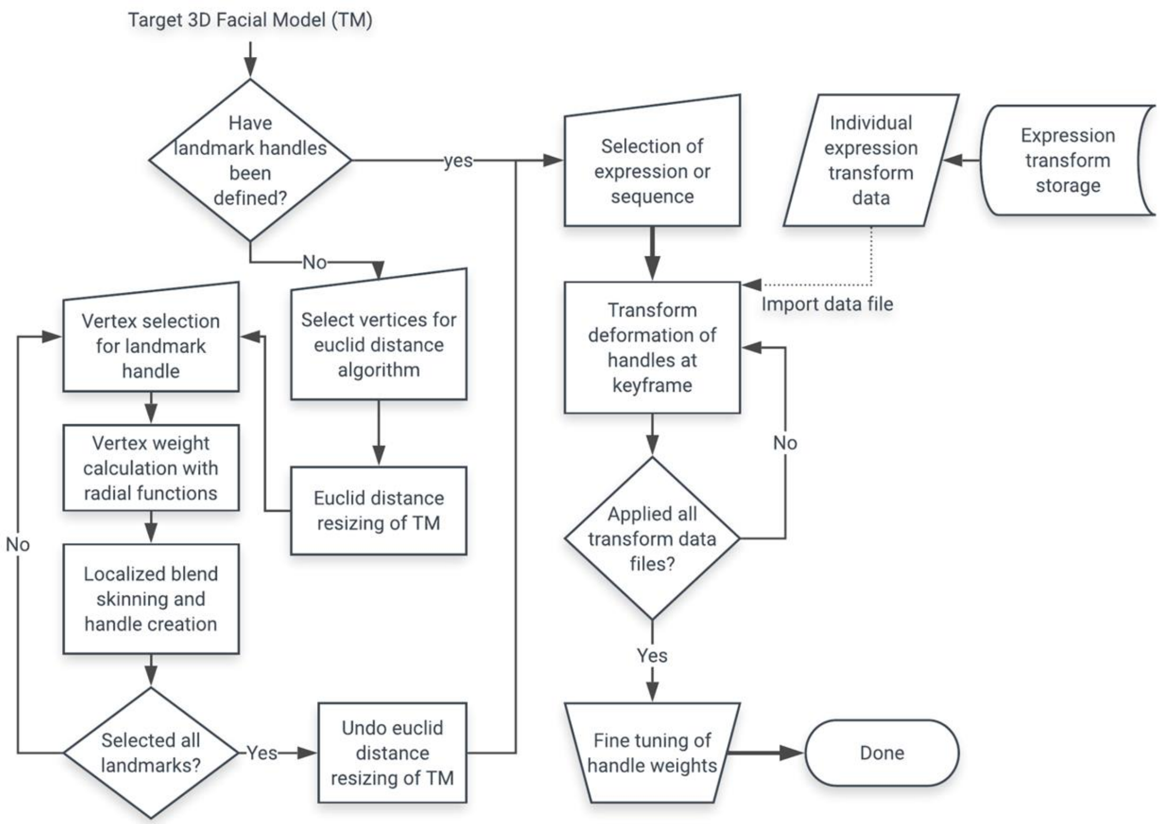


Figure 9: Pipeline for the Weighted Cluster Method

With this first method, we only need to use the actual landmarks and their positions for each pose/animation that was extracted from the base model. For each pose or keyframe in the animation, we create vectors from the difference of the landmarks in the deformed model and the neutral face. These vectors represent the transformations needed to recreate the pose of the deformed model in the target model. We now proceed with the selection of landmarks on the target model along with the method to select the surrounding vertices from these landmarks which will define how landmarks manipulate the translations of the connected vertices.

3.3.1 Radial Basis Function Weights

A radial function is a Euclidean space defined function whose value depends on the distance between a sample point and the origin. A radial function in three dimensions can represented as:

$$\Phi(x, y, z) = \beta(r), \ r = \sqrt{x^2 + y^2 + z^2} \tag{1}$$

Radial Basis Function

Where β represents a function of a single non-negative real variable. In the context of computer graphics, and more specifically for use in the creation of control points for facial feature landmarks, this radial function represents the area of influence around the center point, which in our case is a facial feature landmark.

Now that we have an area of influence defined by a sphere of radius *r*, we can replace β with a Gaussian radial basis function in the form of:

$$\beta(r) = b * e^{-a*(r^2)} \tag{2}$$

Gaussian Radial Basis Function

Where b represents an upper bound for the weight definition, and a corresponds to the shape parameter of function. We set values for this equation; b = 0.75, a = 10, and replacing the power of r with 6 to increase the spread of the initial value, such that:

$$\beta(r) = 0.75 * e^{-10*(x^6)} \tag{3}$$

Gaussian Radial Basis Function with custom parameters

In figure 10 we can see that with these values, the upper bound of weight value (y-axis) will be 0.75; this is done so that the nearest vertices the landmark have almost same weight and the falloff starts after few vertices. If you go higher like weight value 1, then the weight of the landmark would be applied to more vertices and the falloff would start further away. By setting *a = 10* and the power to 6 we make sure that the falloff of the curve will not be as pronounced to begin with but falls off to 0 as we approach 1 on the 'x' axis. If the value of "a" is reduced to 1, then the normalized distance from the center (landmark) to the vertex will be reduced, resulting in a steep curve will not provide good results. We found these values for a, b, and r to be the ideal value for most face models through trial but if it does not work with a certain model, these values are customizable.

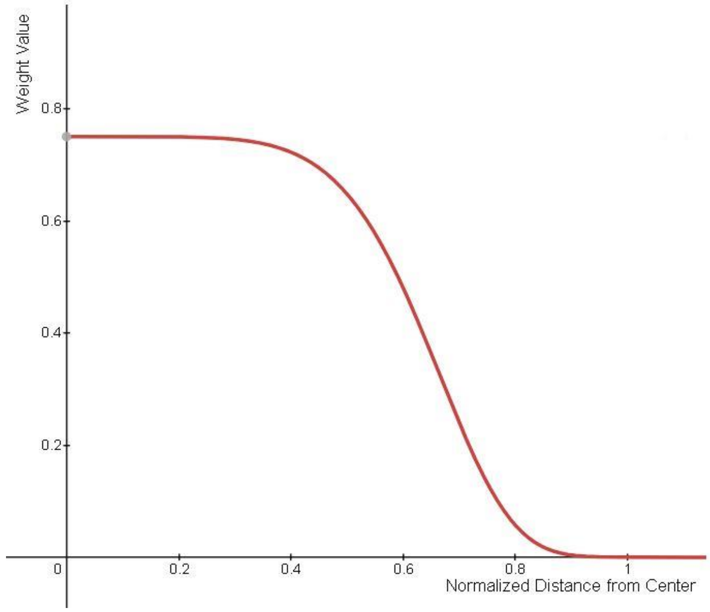


Figure 10: Gaussian Radial Basis Function. Defines the weight value any given vertex should have when located within the influence of the radial function around a landmark

Now we have a method of selecting vertices around landmarks given a radial function with a specific radius, and a radial basis function to determine the weights that vertices within that radius should have, we have to tackle a new issue; all vertices within the volume of the radial function would be selected, this is unwanted behaviour in some cases. What we want instead is to consider the depth of the vertices that fall within the radius. As an example of why this is needed, consider selection of the landmark “Upper Lip Center”, if we don’t take into account the depth of vertices, then the bottom lip vertices would also be selected. To calculate depth, we can make use of Dijkstra's Shortest Path algorithm to calculate the shortest distance through the surface of the mesh to the center point. Using this distance with the radial basis function allows us to overcome this issue and select the appropriate vertices.

3.3.2 Localized Blend Skinning

In regular Linear Blend Skinning, we assign all vertices weights to different bones or handles such that they add up to 1, this is to accomplish rigid transformations of a mesh. When it comes to facial models, we can implement a modified skinning method based on the locality of the landmarks. With this method, all weights on vertices do not have to add up to one, this allows for any given vertex to have a weight lower than 1 and only be affected by one control point. Any vertex could also have multiple weights by being in the influence area of two or more control points. With this in mind, we can assign the vertices and their weights calculated from the radial function and shortest path algorithm to their corresponding landmarks in order to create the control points that will modify the vertex transforms.

This method provides a fast and simple expression transfer system which results in smooth and realistic expressions at the cost of accuracy. This means that some transformation data is lost during the transfer, but the system accounts for this by still providing transformations for the relevant vertices based on the landmark transformations.

3.4 Vertex-based Animation Transfer Method

In the previous section, we see how much of the deformation data from the base model is lost since we only consider the landmarks themselves whilst ignoring the rest of the vertices. This results in smooth and pleasing animations in the target model, and while this improves the speed of a facial animation transfer system, it is not an accurate and true animation transfer between any two face models, it should be used when accuracy is not the main focus, but instead speed and simplicity.

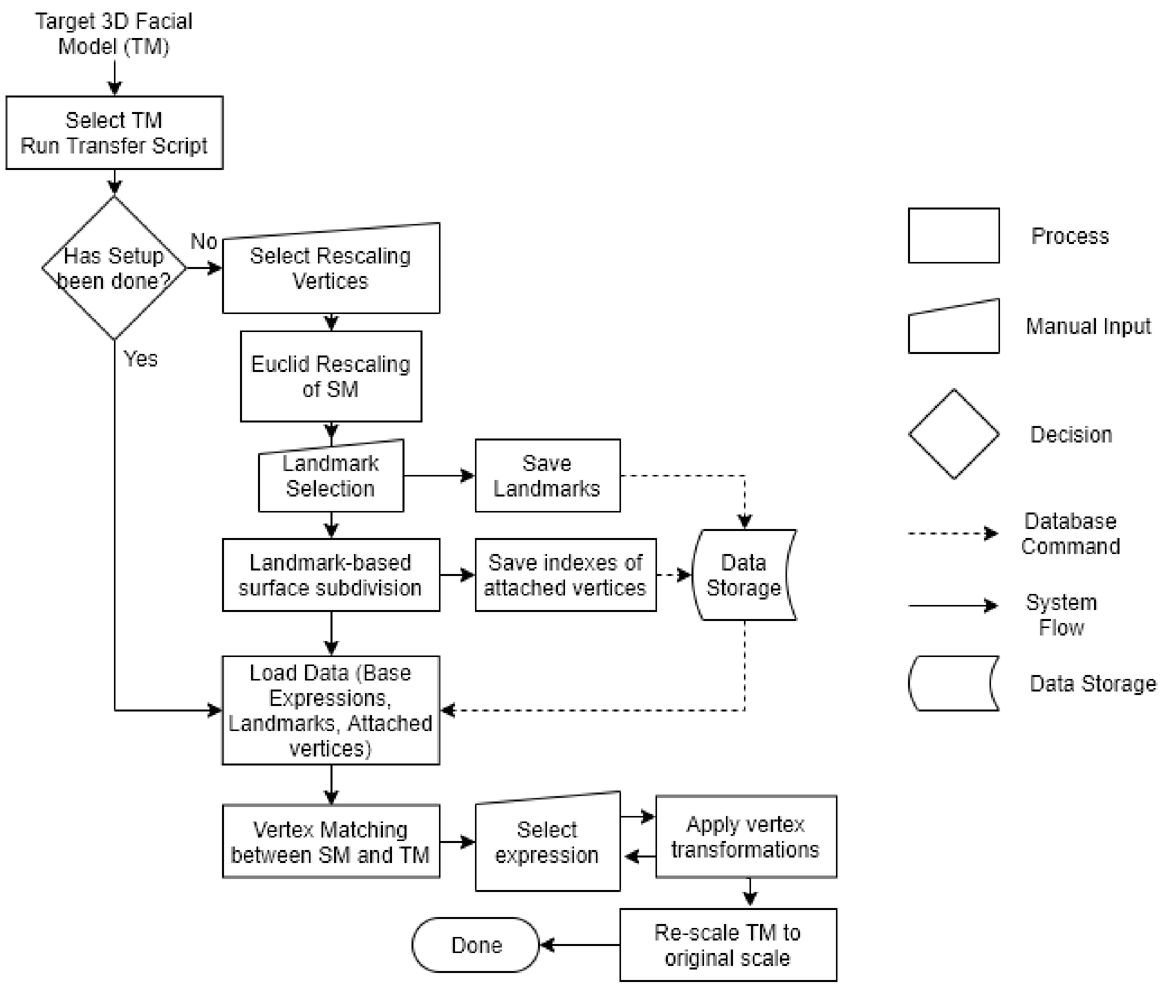


Figure 11 Pipeline describing the manipulation of target 3D facial model and animation application

In figure 11, we depict the application process of the system. To start the user would import a target model into a blank space in Maya. Then they would run the system script. If the user hasn't done the basic setup for the target model, they would have to do that using steps mentioned in the extraction process except the expression extraction step. If the basic setup is done, then the system would load data from base and target model like expressions from base model, and landmarks and attached vertices from both base and target model. Then the system would perform vertex-matching in both models that means for each landmark region on the base model, the landmark in that region would create a vector array for all the vertices in that region. After creation of vector array on base model, we would repeat the process for target model where we would start with an initial vertex to create a vector between the landmark and the vertex in the particular region. Then we would check the distance from the tip of all the vectors from the base model to the tip of this particular vector, out of all the vectors whichever has the shortest distance, the particular vertex on the target model would be attached to that vertex on the base model. This process will carry on till all vertices are matched. Then the user would select a particular expression for the target model. To apply the animation, we would look at all the vertices of the neutral face and all the vertices of the selected expression face on the base model to create a vector for all the vertices from neutral face to its corresponding vertices on the expression face which would create a movement vector. Then the movement vector array is applied to all the vertices of the target model which were matched to all vertices of source model using vertex matching. The selection of expression can take place n-number of times till the user is satisfied with the animation created. The animation can be single, multiple (one after the other), and/or overlapping. After that, the model is rescaled back to its original scale.

In this section we introduce a more comprehensive and detail preserving method that takes into account all vertex transformations from any base model. By using landmarks to divide the base and target mesh into regions, followed by a vertex matching algorithm to match vertices between the two models, and finally analysing and comparing the landmarks between the two models to alter the transformations to be applied, a complete facial animation transfer method is provided.

3.4.1 Landmark Definition

In Chapter 3.3, we talked about how FAPs comes into play in our system. As mentioned, we narrow these facial feature landmarks to a much more manageable 14 facial feature landmarks. Different to the traditional use of landmarks in facial model analysis, where landmarks define certain muscles, we propose an inverse definition where landmarks are strategically chosen, followed by certain animations or poses which will affect the transforms of the landmarks.

Table 6: Expressions and their effect on chosen landmarks

Expression	Landmarks Affected	Example
Eyebrow Raised	1, 2, 3, 4	

Eyes Open Wide	1, 2, 3, 4, 5, 6, 7, 8	
Mouth to Right	9, 10, 11, 12, 13, 14	
Snarl	1, 2, 3, 4, 9, 10, 11, 12, 13, 14	

Smile	9, 10, 12, 13	
Cheek Puffed	9, 10, 12, 13, 14	
Jaw Open	12, 13, 14	

Jaw to Right	13, 14	

To elaborate on the purpose of landmarks, they are not used as control points like the previous method did, instead, they are used as reference points in the analysis of the mesh and vertices for setting up the subdivision and vertex matching as described in the following subsections.

3.4.2 Landmark-Based Surface Subdivision

The process of surface subdivision consists of dividing the mesh into separate regions. In order to do this, we focus on each vertex of the model and find the shortest path from that vertex to each landmark, the shortest path among all of these will set that vertex to the corresponding landmark, making it part of the region. The shortest path is important since it is looking for the distance of a path along the surface of the mesh. If a different method to assign vertices to a landmark such as the Euclidian distance between them is used, erroneous regions will be defined. As an example, let's take the top and bottom lip landmarks. Depending on the initial pose of the mouth, using the Euclidian distance might mistakenly assign vertices on the bottom lip to the top lip since the distance would be shorter due to initial pose of the mouth, and thus creating a region that is not correct. If instead we use shortest path on the surface of the mesh, the shortest path of the top lip to a vertex on the bottom lip would have to go around the corner of the mouth to the right or left before arriving at the destination, significantly increasing the distance compared to the result of a simple Euclidian distance analysis.

After the process is completed, all vertices will be assigned to one of these areas around the landmarks. Below is a representation of the regions in the mesh after they have been assigned to one of the landmarks.

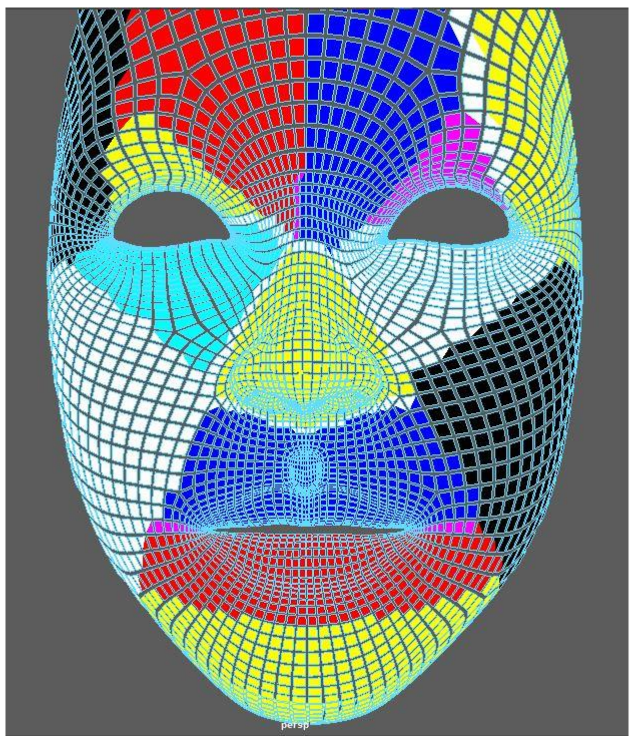


Figure 12: Region definition based on the landmarks

This process is done on both the target model and the base model. Since all we need to perform this step is the landmarks and the position of the vertices, the extracted data from the base model is all we need as well as the equivalent data on the target model. With these regions defined, we can now move on with matching the vertices between the target and the base model.

```
L[] = Selected Landmarks
V[] = Model's Vertices
For each vertex in V:
        shortest_distance = 100
        region = none
        For each landmark in L:
                d = distance_of_shortest_path(vertex, landmark)
                if d < shortest_distance:
                        shortest_distance = d
                        region = landmark.region
```

Figure 13: Pseudo code for region definition

3.4.3 Vertex Matching

The process of vertex matching consists of matching all vertices of the target model to some vertex of the base model. This ensures that the most level of detail is preserved in the transfer since the transformations of all base model vertices are taken into account. Matching vertices, however, is not a simple process. We cannot simply compare the global positions of the vertices since each model would have different topology, vertex count, and vertex ordering. If we overlay one model on top of the other and assign the closest vertices to each other, there would be heavy mismatches since the topology would make different areas drastically misplaced. For example, a longer face would have the mouth position be much lower than a shorter face, meaning the lips would not be matched perfectly, causing errors in the transfer.

To start the vertex matching, we use our landmarks as reference points. We need a way to be able to match vertices depending on the region that they correspond to, and as such, we need a relative comparison between the vertices on the base and target models. Taking a sample vertex ***V*** from a landmark region ***R*** with the corresponding landmark ***L***, we can define a region vector from landmark to a vertex ***R: L -> V***. In this way, we define a set of vectors for each region:

$$R_n : \{L_n \rightarrow V_i\} \tag{4}$$

Definition of regions

Where 'n' is the region number which would be same as the landmark number, and 'i' is the vertex index.

This set of vectors, available on both the base and target model, is used by each vertex on the target model to compare with every vertex of the same region on the base model. This comparison is to find the closest match to this vector. Since we already calculated the vectors so that both vectors are relative to each, we need only to find the closest distance between the heads of the vectors to find the closest match. Once a match is found, the vertex index of the base model is saved for the target model vertex being analysed. This pairing of vertices is what will guide the animation transfer, since it is the closest and most accurate point based on the different topologies.

```
RbVbVector[] = array of vectors from landmark to all vertices in landmark region on base model
RtVtVector[] = array of vectors from landmark to all vertices in landmark region on target model
For each base model vertex Vb in assigned region Rb:
    RbVbVector = Rb -> Vb
For each target model vertex Vt in assigned region Rt:
    RtVtVector = Rt -> Vt
    shortest_distance = 100
    matched_Vb = null
    For each RbVbVector:
        d = distance(RbVbVector, RtVtVector)
        if  (d < shortest_distance):
            shortest_distance = d
            matched_Vb = RbVbVector.Vb
```

Figure 14: Pseudo code for Vertex matching

3.4.4 Geometric Transformation

The first step to apply a transformation taken from a base model is to check the matched vertices of the base with the target model and apply the transform that will take place when an expression or pose is considered. Similar to how we find matches for vertices, we must find the vectors for each vertex on the base model that will define the movement to a pose or expression. By taking the difference between the expression vertices and neutral vertices, we find the transformation vectors that can be applied to the matched vertices on the target model.

In order to assure transformations have proper magnitudes on the target model when it comes to the movement of eyelids is to compare the landmarks themselves. The dimensions of facial components will differ from model to model; this becomes an issue with the eyelids since the calculated transformations may not be enough or may be too much to close the eyes properly. To picture this, we can imagine setting the base model eyelids to be perfectly closed, however, due to differences in how wide eyes are open in the target model, the transformation may not represent a closed eye; if the eyes are open wider on the target model, then the applied transformation would show the eyes closing, but not all the way, vice versa, if the eyes are narrower on the target model, this may cause the eyes to close too much and clip through geometry. This can be accomplished by taking the Euler distance of the top and bottom landmarks of both eyes for both models and comparing their differences to adjust the magnitude of the transformations. The difference is added to the transformation of the vertices in the regions corresponding to the landmarks to guarantee the final transformation is accurate and avoid the issues of geometry clipping or not appropriate transformation calculation.

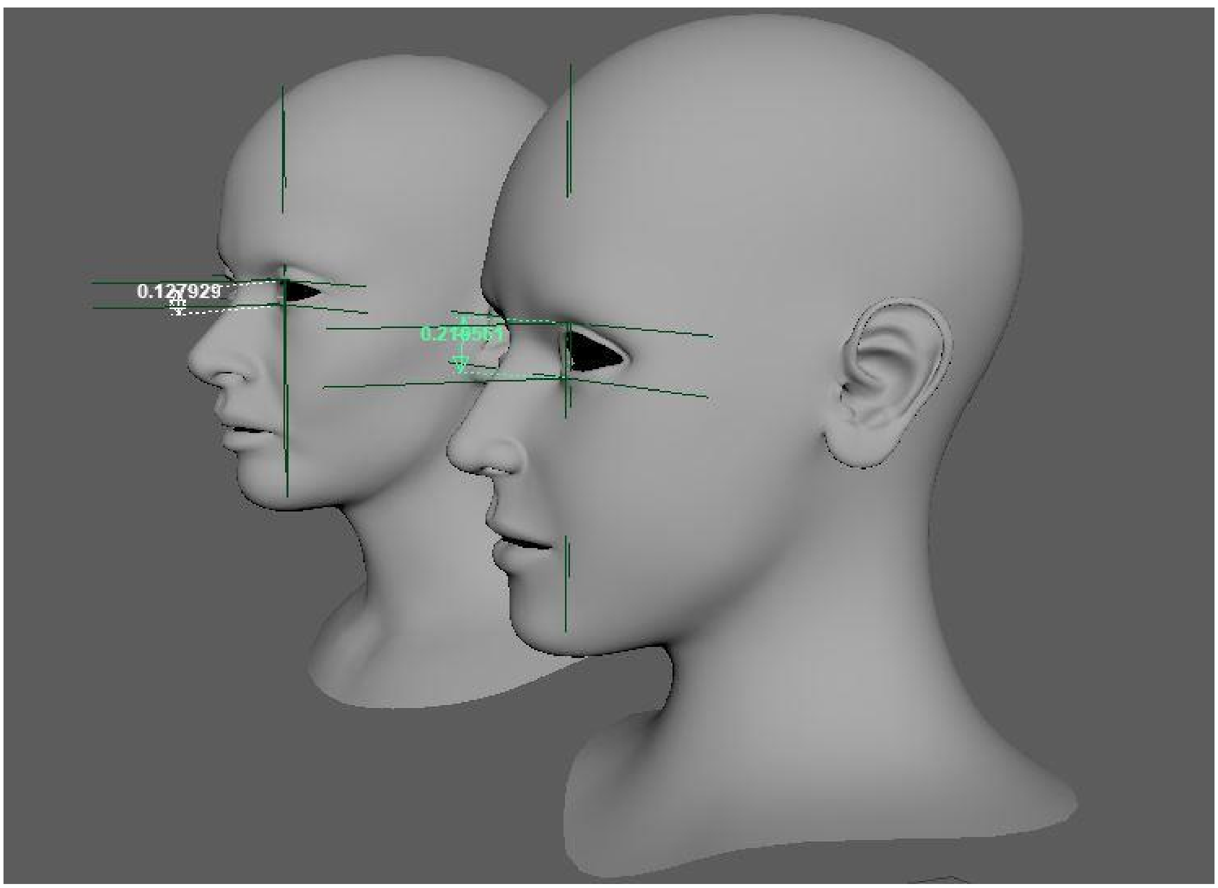


Figure 15: Comparison of eye dimensions between target and base model landmarks

Finally, a smoothing factor is applied to the model by averaging the vertex positions up to a certain depth level. By sampling the mesh to retrieve a vertex, we return the connected vertices up to a depth level specified by the user and average the positions in order to create a smoother result. This step is optional, and it is included for the sole purpose of making the result smoother if the original transfer resulted in any sharp edges or other unwanted artefacts.

3.4.5 Facial Animation Transfer

In this section we provide details of the facial animation transfer. For the current version of the system, a predefined set of pose-to-pose expressions are defined that can be selected from, as well as selection from four animation sequences described earlier. We will start by describing pose-to-pose and move on to the animation sequence.

3.4.5.1 Pose-to-Pose Animation Transfer

For pose-to-pose animation transfer, a set of individual expressions are extracted from the source model sequences and stored as individual transformation data files. These can then be selected as individual expression transfers into any target model. With the current landmarks defined, the system continues to load the individual transformation data file that was selected and apply the vector transformations from each matched vertex on the base model to the target model. Since it is an individual expression that is been applied, the timestamp for the keyframe is ignored, instead, the user is free to assign a certain time to insert a new keyframe with the new expression applied. This process is repeated indefinitely until the user is satisfied with the number of poses applied to the animation. This method is meant to be used as a standalone expression application. For the purpose of this work, poses are extracted from a base model, essentially making it a transfer method, however, this process can be refined to include general facial expressions that can be extracted or crafted manually for ready off the shelf facial animation creation on any face model. This minimizes the workload on animators looking to create sets of facial expression animations.

3.4.5.2 Animation Sequence Transfer

Similar to how pose-to-pose animation transfer is performed, an entire extracted sequence of animations can be selected. This is meant for use with more custom and specialized facial animations; instead of simple expressions such as smile, frown, blink... Given a model with an animation sequence, the system extracts all the deformations defined in each keyframe and continues to apply the entire animation to the target model using the methods described above. Using the keyframe timestamp, the system knows when to apply each expression to the target model, resulting in a one-to-one copy of the sequence from the base to target model. This method is flexible, allowing the user to change the animation sequence in some way, by adding in-between keyframes with different expressions and reapplying the animation to the target model without having to go through the initial setup of selecting landmarks.

3.5 Model Setup

The goal of the system is implementing an automated facial animation system, which is able to quickly and efficiently create smooth and realistic facial animations based on deformation transfer between the selected expressions. This section would elaborate on how the whole process happens and what is viewed by the user.

To start the process, the user has to import or create a custom target face model in Maya and run the script called 'Automatic_Facial_Expression.py'. Now a welcome window will open up asking the user to select the model they want to work with i.e., either base or target model.

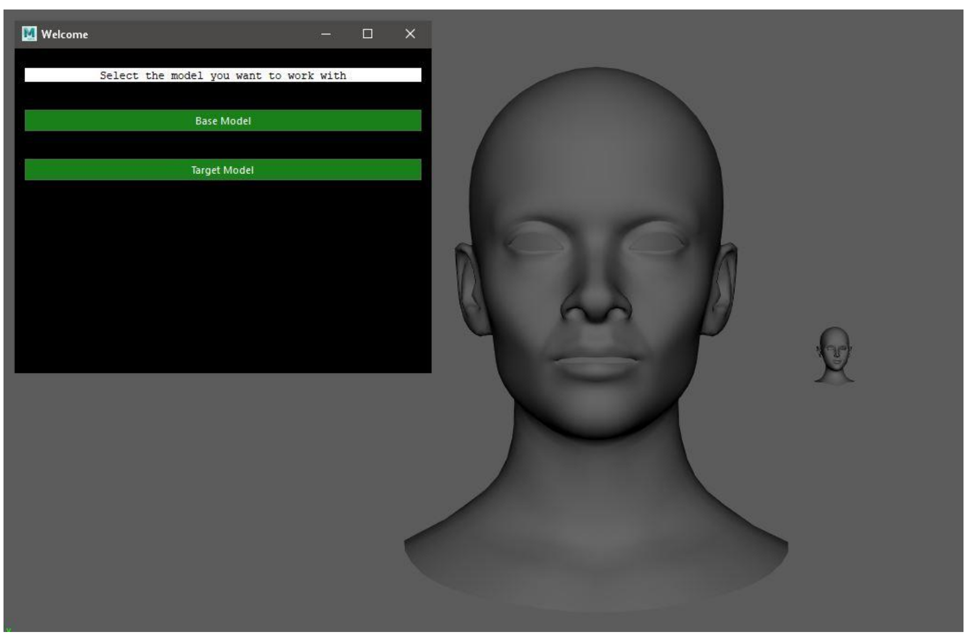


Figure 16: Welcome and model selection window

The second window will ask the user to select the model they choose in the previous window in the object model. This will allow the system to register the name of the face model imported which is needed for different parts in the algorithm. Now if the user had selected "Base model" in the previous window, they would have to choose between "Landmark Selection" or "Animation Extraction" as shown in figure 19. But if the user chose "Target Model" in the previous window, then they would have to choose between "Landmark Selection" or "Animation Application" as shown in figure 20. The first option will be used for a new target face model to start the whole process for the landmark selection or to edit any pre-existing landmarks that the user is unsatisfied with. The second option is used for a face model which has gone through the process of landmark selection and the user only wants to add/modify the facial animation. When choosing the first option, the script will run in full in order to create the landmarks necessary for facial expression transfer, otherwise, if the second option is chosen, the script skips to the last window where facial expressions are set since it will already be able to reference the landmarks that were created at some point.

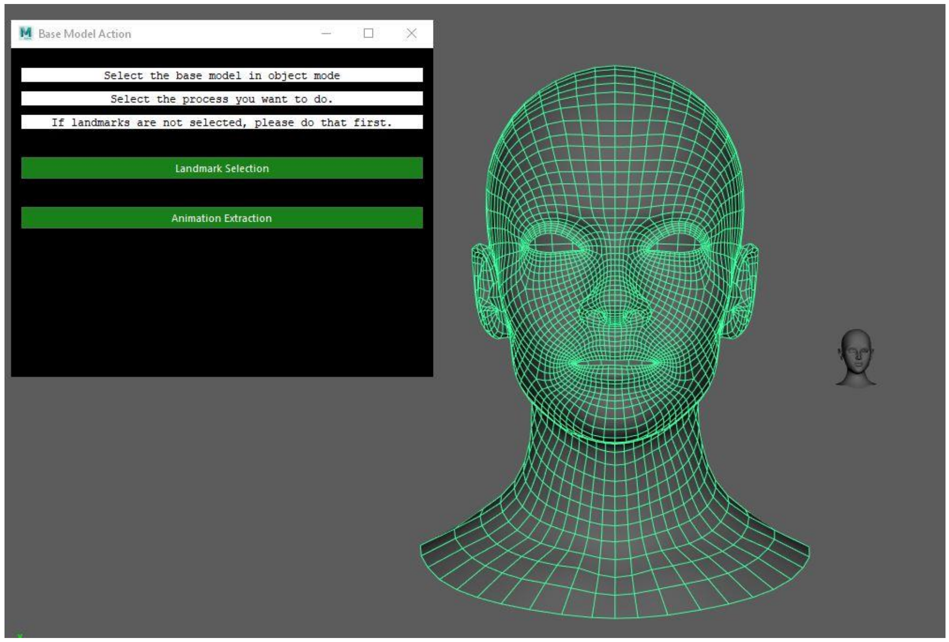


Figure 17: Base model action selection window between landmark creation and animation extraction

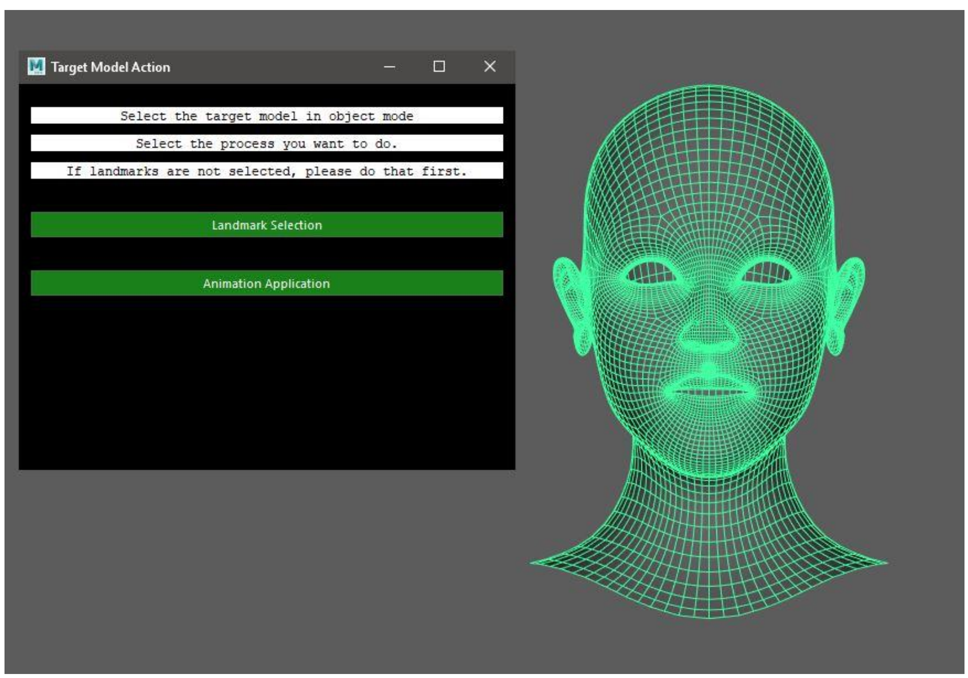


Figure 18: Target model action selection window between landmark creation and animation application

Now this third window will show up if you had selected “Landmark Selection” for the previous window. It will ask the user to change from object mode to vertex mode and select the two landmarks, one close to the forehead and the other close to the chin of the model as depicted in the guide picture. This step will provide the system with the height of the model. After selecting the two points, the user should click “Continue” to move onto the next window. This triggers the resizing function describe in the previous section.

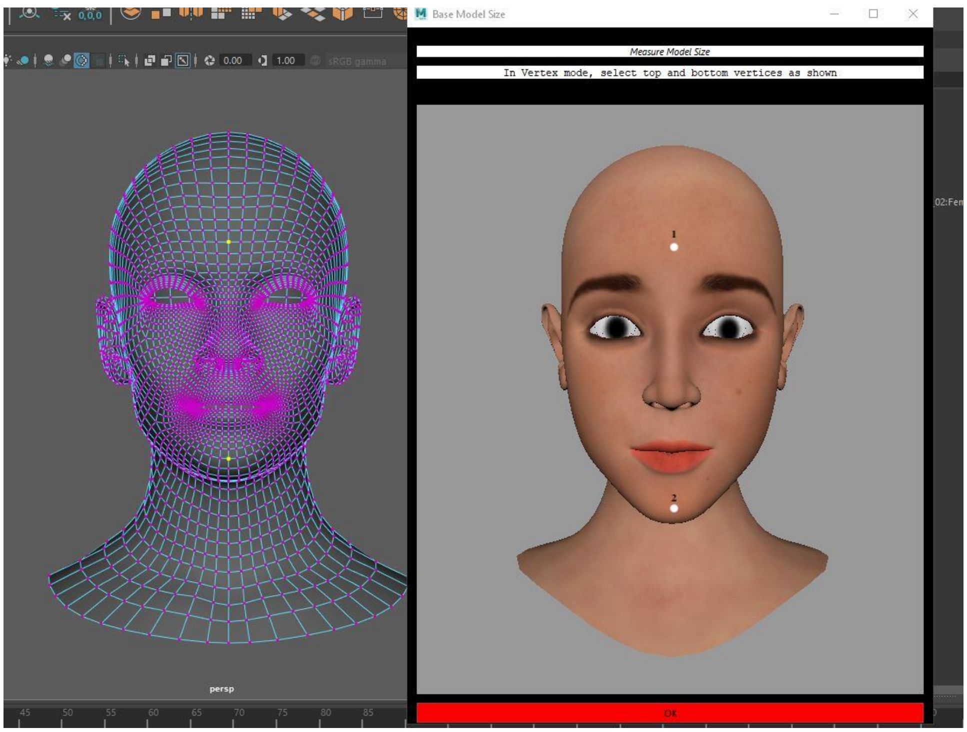


Figure 19: Selection of vertices for model rescaling

The fourth window will be executed only after the previous window and it will be not be called by the “Animation Extraction” option. This window will ask the user to select a certain landmark on the face model as per the guide image (the point doesn’t have to be exact, but as close to the indicated vertex or vertices as possible). The landmarks selected in this window will be recorded and saved so that they can be used with any animation modification done to the base model without the reselection of landmarks. As explained in the system explanation section, the landmarks will allow the method to find the matched vertex for each vertex between the base and the target model. After selecting the landmarks, the user should click "Continue" to start the animation extraction or animation application process, depending on which model was chosen on the first window.

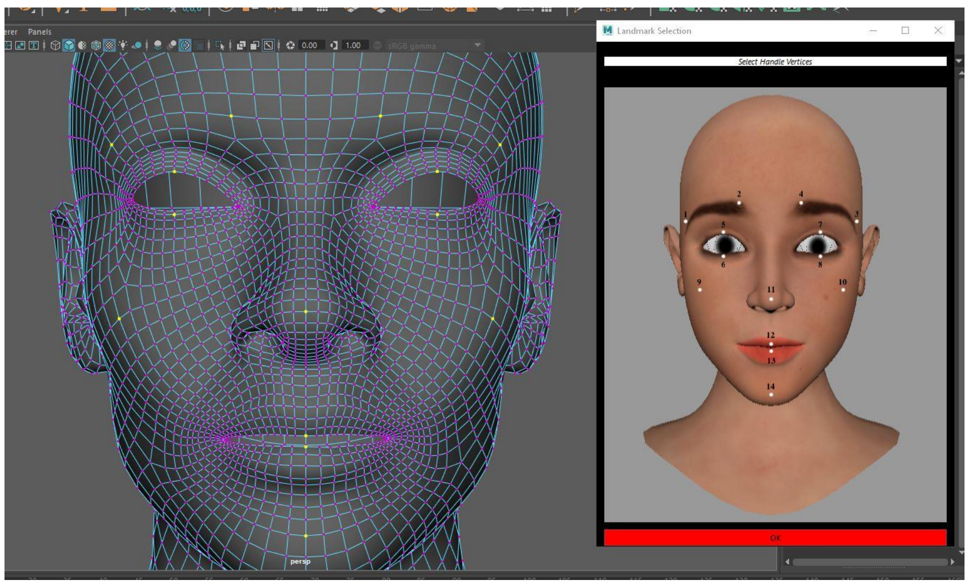


Figure 20: Landmark selection for the model

Finally, if the user selected "Base Model" in the first window, they can either continue onto the target model to perform similar process as mentioned above or exit the window. Whereas, if they selected "Target Model" in the first window, the next window will be specific to what type of animation has been extracted from the base model. For example, if the extracted animation is a manual keyframe animation, the application process would be automating either after selecting the "Animation Application" option or after selecting the landmarks on the target face model. If the extracted animation is using blendshapes for each pose, then you after completing the above-mentioned step, you would be directed to the following window to enter the start and end time for an animation and by selecting the titled pose option, you can apply it. The animations can be individual, multiple (one after the other), or overlapping.

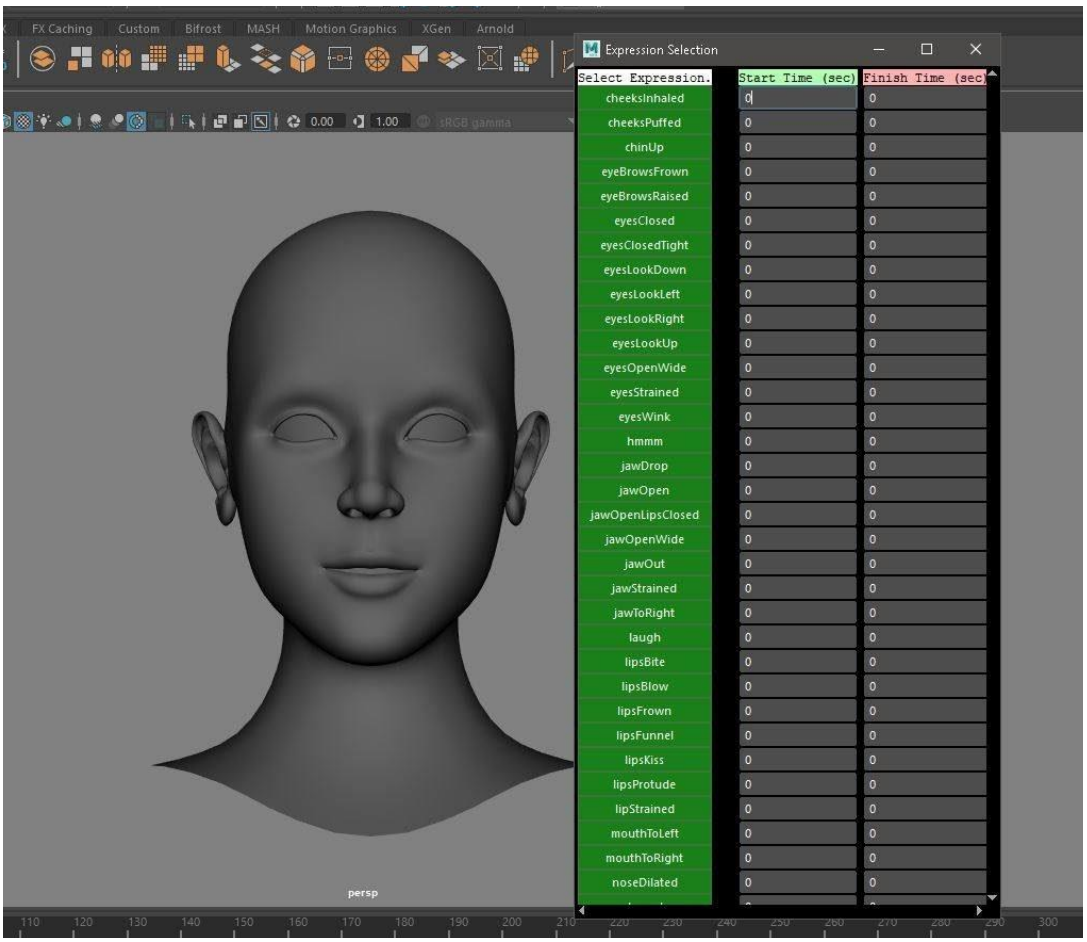


Figure 21: Selection with poses along with their start and end time

The resultant animation is smooth, realistic, and highly defined. After selecting a pose, the window stays open so that the user can select another pose to be appended to the animation. The user can also reset any changes, and finally, apply changes to close the window and finish the whole process.

Later, if the user wants to make another animation with the same model, the user doesn't have to go through all the steps, whereas they can skip to the animation extraction and then go to the animation application steps.

Chapter 4: Result

As stated in previous chapters, the base model has facial expression animation blendshapes animated by a professional animator/modeller. These animations include an individual as well as sequence of expression animations. There are four sequences of animations which consist of different facial expression animations for different amount of time. Sequence 2, 3, and 4 are around ~6-second-long with approximately 5-6 basic expression animations. Sequence 1 is around ~15-second-long with approximately a combination of 45-50 basic and extreme expression animations. The individual expressions are extracted from these sequences to create a list of basic and extreme facial expression animations that can be applied to the target model. Few examples of the individual as well as sequence expression animation are displayed graphically below. To show the degree of accuracy for the transfer of animation, where left-side model is the base model and the right-side model is the target model.

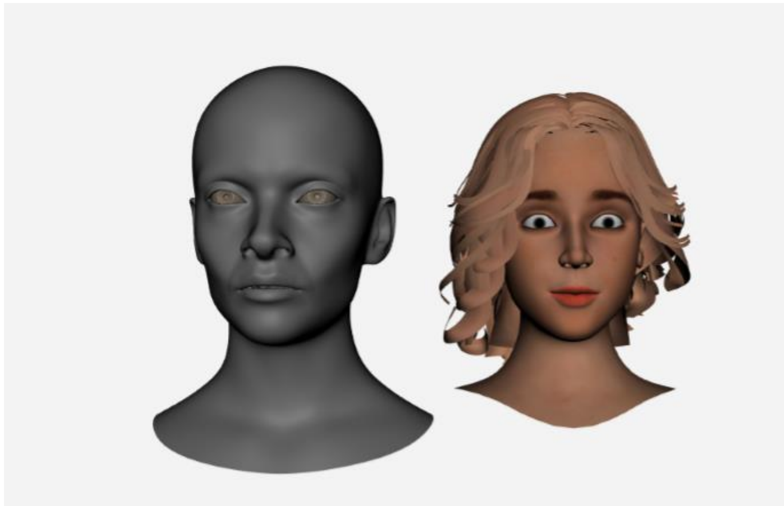


Figure 22: Neutral face for both base model and target model 1

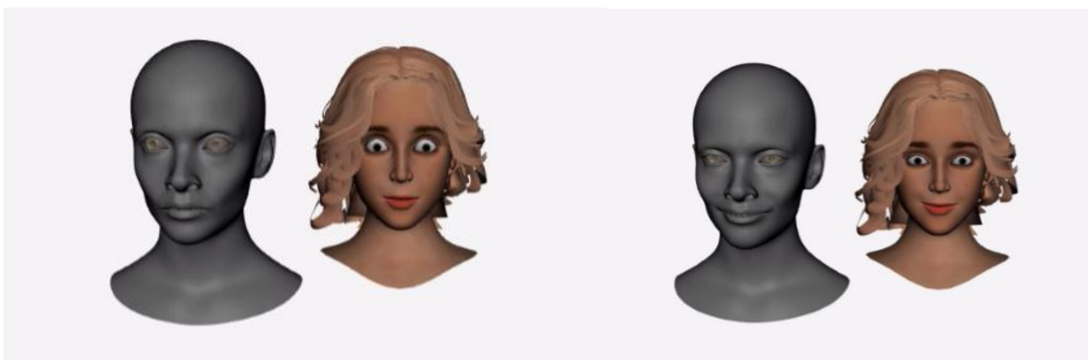


Figure 23: Left - Eyes Wide Open for both base and target model 1. Right - Smile for both base and target model 1.

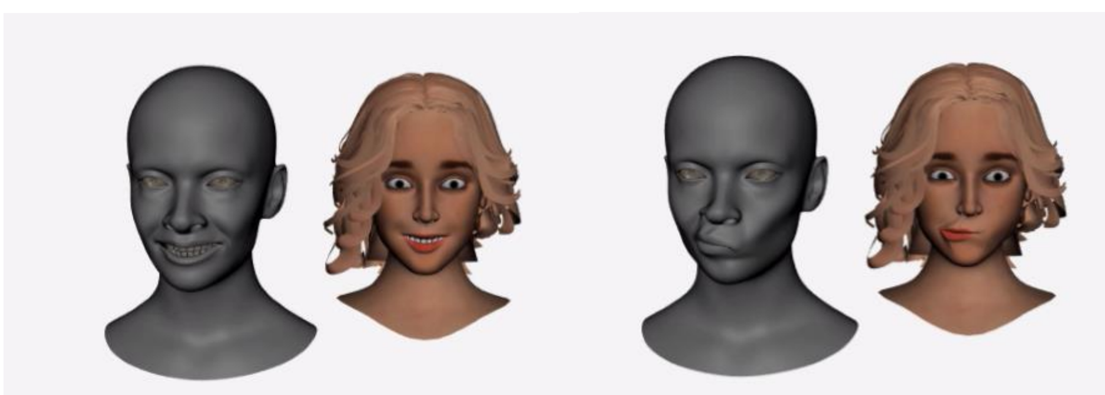


Figure 24: Left - Mild Laugh for both base and target model 1. Right - Mouth move to right for both base and target model 1.

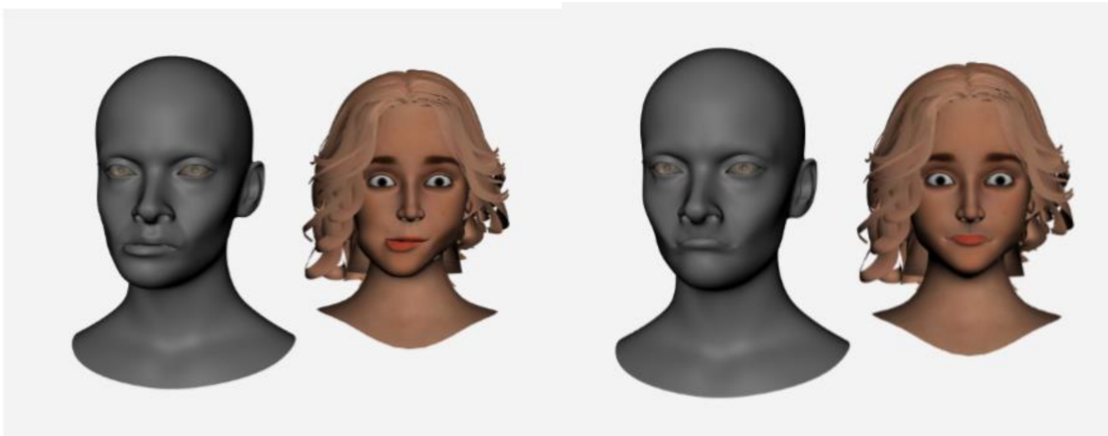


Figure 25: Left - Jaw move to right for both base and target model 1. Right - Lip Bite for both base and target model 1.

Figure 26: Left - Eyes Wide Open and Jaw Open for both base and target model 2. Right - Eyes closed and Lip Bite for both base and target model 2.

Figure 27: Left - Eyes Wide Open for both base and target model 2. Right - Jaw move to right for both base and target model 3.

Figure 28: Left - Mouth move to right for both base and target model 3. Right - Smile for both base and target model 3.

More snapshot can be found in the Appendix; these include individual pose-to-pose facial animation as well as the sequence of facial animation.

The efficiency of the proposed system can be evaluated by comparing it with the process of blendshapes or even other face animation methods, including the time taken for the setup as well as the creation of the animation. Initial setup for blendshapes requires the user to create multiple copies of the base face model uploaded while making sure that the vertex order, vertex count and topology of the base model as well as all its copies is similar, otherwise the process wouldn't work properly. After creation of the multiple copies, the user has to create individual facial expressions on each of these copies before moving onto the process of creating animations using blendshapes. The time taken for this process varies as per the complexity of the model, experience of the user, and so on. Whereas, in case of the proposed system the user needs to upload any 3D face model (with similar or different vertex count, vertex order, and/or topology) and spend around 2-3 minutes to set up the landmarks for the face model, and then they would be able to select and create any of the listed expressions without repeating the setup process for the same face model again. When comparing both the techniques with respect to the setup process, the proposed system would require less time as compared to blendshapes, and it would be able to work on any human type face model, regardless of their topology.

4.1 Perspective from Professional Animators

This sub-section constitutes a few opinions from professional animators and modellers point of view. The following question is asked,

Q. How much time would you say it takes you, more or less, to create a facial animation?

> *It really depends on how complicated the face and expressions are. If the sequence is very simple without exaggerated features, then it would approximately 1 hour or so. But the time highly depends on the model and what you are animating.*
>
> - Raechel Padua, 3D Modeller at Carleton Immersive Media Studios.
>
> *Depending on how big the rig is and how many poses you need to need to make, it can take a few hours.*
>
> - Samy Fecih, Senior Animator at Ellipse Studio.

Depending on the duration of sequence, the creation of expression can take over a day.

- Raul Fregoso, Supervisor of 3D animation at Mighty Studio.

For the current workflow for creating facial animation using Blendshapes, it can be a tedious process which can take anywhere from a few minutes to hours to create a realistic animation.

- Juan Rincon, Animation Student at Ringling Studio.

4.2 Advantages

The proposed system provides a lot of contribution to the existing research and it is addressed as follows.

1. This method preserves and utilizes all the data from both the face model to ensure a smooth and realistic facial animation transfer.
2. This method does not rely on any rig or skeleton of any sort to transfer expression between two different models with completely different topology. A lot of methods require either some vertex order similarity or some sort of rig.

4.3 Limitations

1. The proposed system might not reflect very complex facial modifications done on the base model towards the target model, such as pronounced wrinkles or change in topology. This is specifically more pronounced when transferring expressions

from high resolution models with a high number of vertices to simpler target models with fewer numbers of vertices.

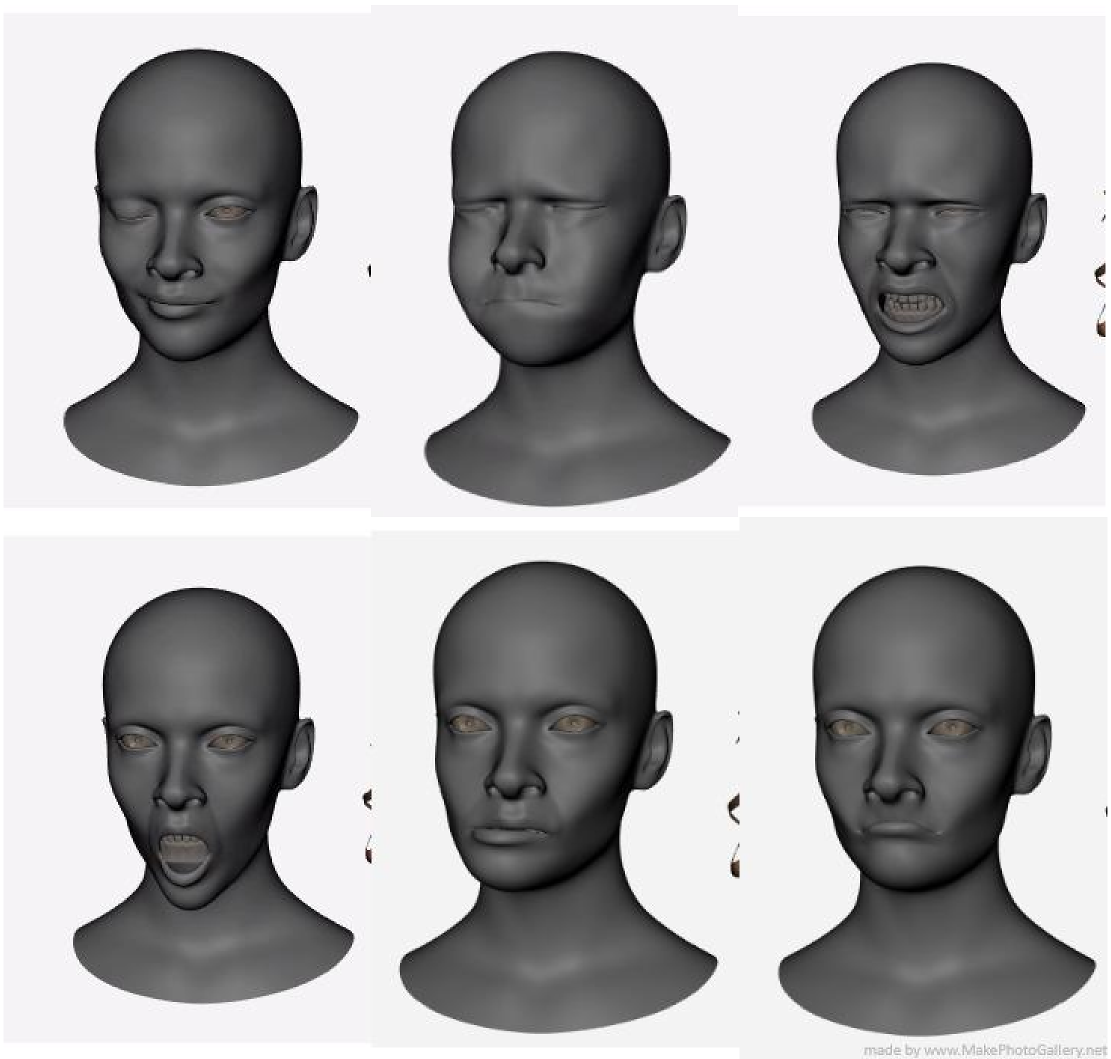
2. The proposed system is only applicable to a facial mesh and not to the separate meshes like eyeballs, teeth, tongue, and hair. This is due to how different these facial components are handled, for example, the teeth are rigid meaning they don't deform in the same way as the facial mesh, and the tongue, eyeball, and hair have unique behaviours as well. This limitation will be addressed in the future with extensive research and implementation of concept.

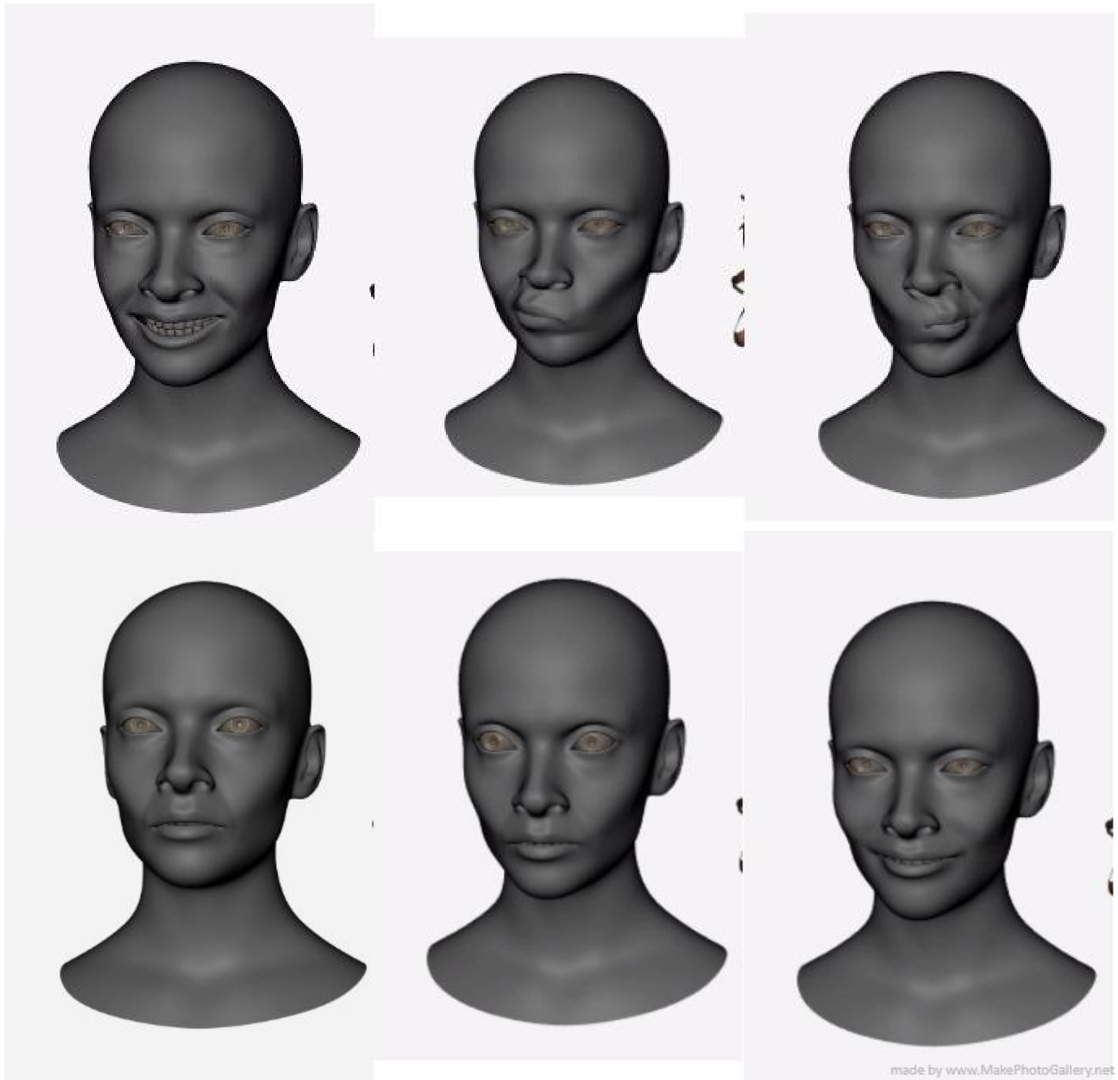
Chapter 5: Conclusion & Future Work

We present a 3D facial animation transfer system between one source face models and reuse them on many different target face models, regardless of their topology. This method allows any sort of animation method as a source face model's animation. The system is purely vertex-based; this helps to speed up the process since no rigging is necessary. Furthermore, the system reduces the amount of manual effort in order to accomplish a smooth and realistic facial animation. The automation of selection process for the landmarks, along with the creation of facial movements database associated with speech will be extensively research and implemented in the future. Another key component to make a system such as this more efficient and faster when it comes to processing information about a large number of vertices is some sort of parallel processing. The main goal of this work was accuracy of animation transfer, however, by researching more into parallel processing to analyse vertices in a more effective way would speed up the system dramatically. Systems to be able to handle all components a face models are another area of research to be looked into; how the eyes, mouth, hair, and other components behave and interact with the main facial mesh. And finally, a big breakthrough for 3D face model analysis comes in the form of automatic detection of landmarks. By combining some of the algorithms used in face detection and computer vision and modifying those to work in a 3D modeling software environment, completely automatic facial animation and transfer could come one big step closer to reality.

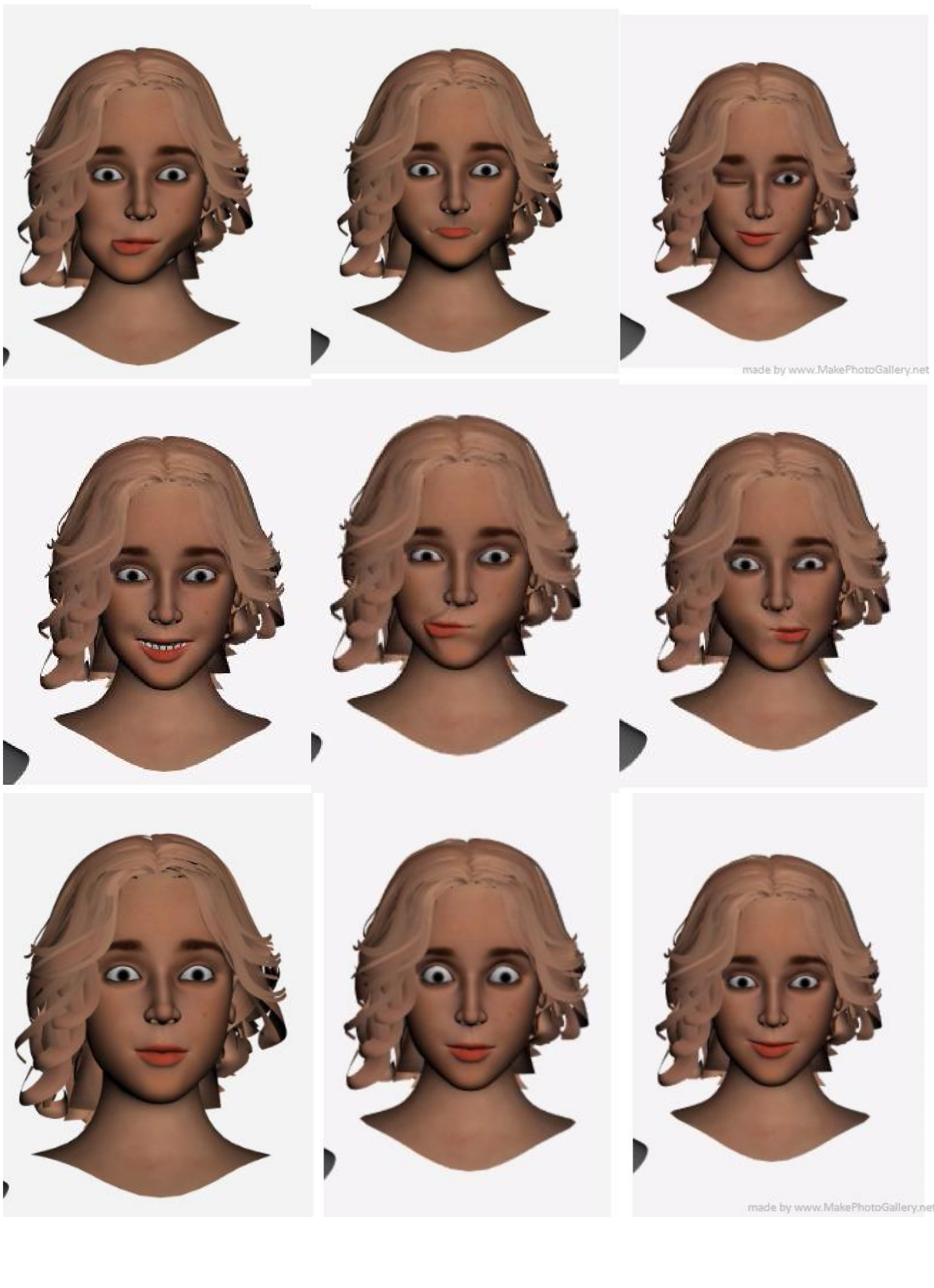
Appendices

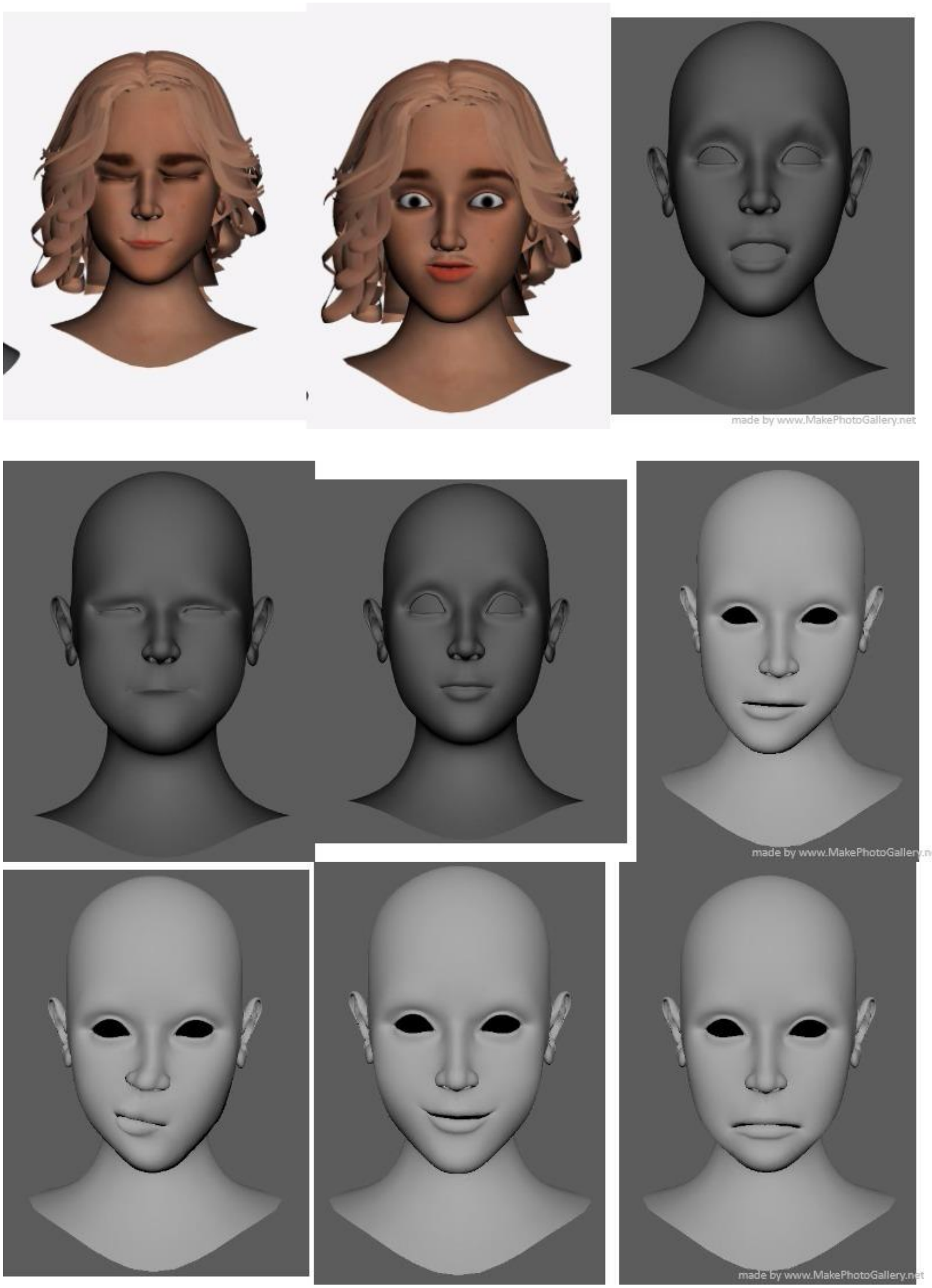
Appendix A – Base Model




made by www.MakePhotoGallery.net

Appendix B – Target Models




made by www.MakePhotoGallery.net
made by www.MakePhotoGallery.net
made by www.MakePhotoGallery.net

References

[1] E. Chuang and C. Bregler. Performance Driven Facial Animation using Blendshape Interpolation. *Computer Science Technical Report, Stanford University*, vol. 2, 2002.

[2] J.X. Chai, J. Xiao, and J. Hodgins. Vision-based Control of 3D facial Animation. *SIGGRAPH*, 2003.

[3] C. Curio, M. Breidt, M. Kleiner, Q.C. Vuong, M.A. Giese, and H.H. B¨ulthoff. Semantic 3d motion retargeting for facial animation. *APGV '06 Proceedings of the 3rd symposium on Applied perception in graphics and visualization*: pp. 77-84, 2006.

[4] L. Dutreve, A. Meyer, and S. Bouakaz. Feature points based facial animation retargeting. *VRST*, 2008.

[5] J. Song, B. Choi, Y. Seol, and J. Noh. Characteristic facial retargeting. *Computer Animation Virtual Worlds*, vol. 22, pp. 187–194, 2011.

[6] F. Xu, J. Chai, J. Lu, and X. Tong. Controllable High-fidelity Facial Performance Transfer. *ACM Transactions on Graphics*, vol. 33, no. 4, pp. 42-50, 2014.

[7] R.B. Ribera, E. Zell, J.P. Lewis, J. Noh, and M. Botsch. Facial Retargeting with Automatic Range of Motion Alignment. *ACM Transactions on Graphics*, vol. 36, pp. 4-10, 2017.

[8] P. Kalra, A. Mangili, N.M. Thalmann, and D. Thalmann. Simulation of Facial Muscle Actions Based on Rational Free Form Deformations. *EUROGRAPHICS*, vol. 11, pp. 3-9, 1992.

[9] X. Lu and A.K. Jain. Deforming Modeling for Robust 3D face matching. *IEEE transactions on pattern analysis and machine intelligence,* vol. 30, no. 8, pp. 1346-1356, 2008.

[10] R. Amini and C. Lisetti. HapFACS: An Open Source API/Software to Generate FACS-Based Expressions for ECAs Animation and for Corpus Generation. *IEEE*, 2013.

[11] C. Cao, Y. Weng, S. Lin, and K. Zhou. 3D Shape Regression for Real-time Facial Animation. *ACM Transactions on Graphics*, vol. 32, no. 4, pp. 41-52, 2013.

[12] S. Agianpuye and J.L. Minoi. 3D Facial Expression Synthesis: A Survey. *8th International Conference on Information Technology in Asia (CITA)*, 2013.

[13] S. Kshirsagar, S. Garchery, and N.M. Thalmann. Feature point-based mesh deformation applied to MPEG-4 facial animation. *Deformable Avatars, Springer*, pp. 24-34, 2001.

[14] J. Ostermann. Face Animation in MPEG-4. *MPEG-4 Facial Animation: The Standard, Implementation and Applications,* pp. 17-39, 2002.

[15] P. Hong, Z. Wen, and T.S. Huang. Real-Time Speech-Driven Face Animation. *MPEG-4 Facial Animation: The Standard, Implementation and Applications,* pp. 115-124, 2002.

[16] N. Tsapatsoulis, A. Raouzaiou, S. Kollias, R. Cowie, and E. Douglas-Cowie. Emotion Recognition and Synthesis Based on MPEG-4 FAPs. *MPEG-4 Facial Animation: The Standard, Implementation and Applications,* pp. 141-167, 2002.

[17] E. Petajan. MPEG-4 Face Animation Conformance. *MPEG-4 Facial Animation: The Standard, Implementation and Applications,* pp. 57-61, 2003.

[18] Y. Zhang, Q. Ji, Z. Zhu, and B. Yi. Dynamic Facial Expression Analysis and Synthesis With MPEG-4 Facial Animation Parameters. *IEEE Transaction on circuit and systems for video technology,* vol. 18, no. 10, pp. 1383-1396, 2008.

[19] Y. Wang, X. Huang, C. Lee, S. Zhang, Z. Li, D. Samaras, D.N. Metaxas, A.M. Elgammal, and P. Huang. High Resolution Acquisition, Learning and Transfer of Dynamic 3D Facial Expressions. *Computer Graph. Forum,* vol. 23, pp. 677-686, 2004.

[20] D. Vlasic, M. Brand, H. Pfister, and J. Popovic. Face transfer with multilinear models. *SIGGRAPH*, 2005.

[21] Y. Yang, N. Zheng, Y. Liu, S. Du, Y. Su, and Y. Nishio. Expression transfer for facial sketch animation. *Signal Processing,* vol. 91, pp. 2465-2477, 2011.

[22] J.L. Minoi, S.H. Amin, C.E. Thomaz, and D.F. Gillies. Synthesizing Realistic Expressions in 3D Face Data Sets. *IEEE*, 2008.

[23] D. Huang and F. De La Torre. Facial Action Transfer with Personalized Bilinear Regression. *ECCV*, ver. 2, pp. 144-158, 2012.

[24] C. Pawaskar, W.C. Mat, K. Carnegie, J.P. Lewis, and T. Rhe. Expression Transfer: A System to build 3D Blendshapes for Facial Animation. *28th International Conference on Image and Vision Computing*, pp. 154-159, 2013.

[25] P. Eisert. MPEG-4 Facial Animation in Video Analysis and Synthesis. *International Journal of Imaging Systems and Technology*, pp. 100-110, 2003

[26] J. Hamm, C. G. Kohler, R. Gur, and R. Verma. Automated Facial Action Coding System for Dynamic Analysis of Facial Expressions in Neuropsychiatric Disorders. *Journal of neuroscience methods*, vol. 200, pp. 237-256, 2011.

[27] M. Zollhöfer, M. Martinek, G. Greiner, M. Stamminger, and J. Süßmuth. Automatic reconstruction of personalized avatars from 3D face scans. *Journal of Visualization and Computer Animation*, 2011.

[28] T. Weise, S. Bouaziz, H. Li, and M. Pauly. Real-time performance-based facial animation. *ACM SIGGRAPH*, vol. 30, no. 4, pp. 77-88, 2011.

[29] V. C. T. Orvalho. Reusable Facial Rigging and Animation: Create Once, Use Many, Ph.D. dissertation, Dept. of Mathematics, Universitat Politècnica de Catalunya (UPC), Barcelona, 2007.

[30] Y. Zhang, and W. Wei. A realistic dynamic facial expression transfer method. Neurocomputing, vol. 89, pp. 21–29, 2012.

[31] S. G. Gunanto, M. Hariadi, E. M. Yuniarno, and M. B. Nendya. Facial Animation of Life-Like Avatar based on Feature Point Cluster. Journal of Engineering Science and Technology Review, vol. 10, no. 1, pp. 168-172, 2017.

[32] B. Bickel, M. Lang, M. Botsch, M. A. Otaduy, and M. Gross. Pose-Space Animation and Transfer of Facial Details. ACM SIGGRAPH Symposium on Computer Animation, pp. 57-66, 2008.

[33] S. Bouaziz, Y. Wang, and M. Pauly. Online Modeling for Real-time Facial Animation. ACM Trans. Graph., vol. 32, no. 4, article 40, 2013.

[34] V. C. T. Orvalho, E. Zacur, and A. Susin. Transferring Facial Expressions to Different Face Models. Ibero-American Symposium on Computer Graphics, 2006.

[35] J. V. Bouguet. Pyramidal implementation of the Lucas Kanade Feature Tracker. 1999.