Perception-Motivated High Quality Stylization

by

Hua Li

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

Doctor of Philosophy

in

Computer Science

Carleton University
Ottawa, Ontario

Table of Contents

Abstract

Non-photorealistic rendering (NPR) algorithms are used to produce stylized images, e.g., in a painted or stippled style. This thesis presents a new family of automatic methods for effects and styles including halftoning, screening, stippling, line art, and mosaics. Our proposed algorithms are motivated by perceptual effects including contrast and similarity to present structure preservation. Meanwhile, the novelty also shows in exploring primitives in distribution for our NPR algorithms.

First, we propose a novel contrast-aware error diffusion algorithm based on a priority-based scheme, called Prioritized Contrast-based Error Diffusion (CED), to generate stylized imagery. The core idea of this technique is to implicitly and progressively preserve the original tendency in contrast. This method generates halftoning with a success in similarities for tone, structure, and contrast, and with good visual appearance. We extend Prioritized CED with variations on masks and priority configuration to create screening style with non-uniform patterns. To apply Prioritized CED to stippling generation, we design a new modification to the contrast-aware error distribution to provide density control in dot distribution. A set of varied stippling styles are produced. Thanks to the good structure preservation, all resulting effects are quite good. We also propose a technique of pixel clustering and employ skeletonization for generation of line art with long strokes. Pixel clustering produces long lines by processing a set of pixels, which align along the edge guidance from a smoothed bilateral filter on an edge tangent field (ETF). Skeletonization captures large-scale structure of an image. We can create simplified line art with clean and elegant visual appearance.

Second, we propose a novel automatic approach to construct an artistic tessellation (AT) through the growth of curves in a particle system. The core idea is to present similarity, and in addition, to consider spatial balance during curve growth. The AT method is applied to create natural and abstract patterns. The further exploration guided by a bilateral smoothed ETF from an image can simulate stained-glass mosaics.

Acknowledgements

It is a long long journey towards obtaining my doctoral degree in Computer Science. This thesis would not have been possible without the contributions and support of my family, friends, and peers.

First and foremost, I would like to thank my big happy family. My beloved husband, Fenghua Shang, is the fundamental role for me to make the decision of studying in Canada. Thanking him doesn't seem to be enough, there are no words which can cover how much I enjoy to be with him, accompanying obtaining my final degree, as well as all of the challenges, new home and new country, and valuing my ideas. His strong support cleans my hesitation on my academic decision. Without any doubt, I am the luckiest woman in the world. I would also like to thank my son Eric, who makes me grow up and rediscover my ability to handle all of the changes. When he smiles, I feel that the whole world smiles! I am deeply indebted to my parents for their unconditional love, support and encouragement. I thank my dad JiAn Li. I am so lucky inheriting some portion of his wisdom and courage. When I talk to him, any confusion is gone and the goal is clearly right there. I thank my mom Jing Huang. I always want to be a strong mom like her to handle so many kids while teaching them to have passion for life. I thank my siblings, Jian, Kang, Wen, and Xia, my best friends. Without their care and enjoyment of life, it couldn't make this thesis happen. I miss you so much. No matter where I am, I am with you all always.

Secondly, but not less importantly, I am deeply indebted to my advisor Professor David Mould for his guidance and encouragement and would like to express my profound respect for him, who has worked shoulder to shoulder with me along this long passage. His boundless inspiration has a huge impact on how I approach the research and his enthusiasm played a key role in achieving good quality research results. I am so fortunate to work with him. He believed in my ideas, gave me the freedom to pursue my goals, and supported me generously wherever he could. Especially, when we laughed at the naïve ideas, his insightful thoughts can always find the pieces of

treasures from them, which makes this thesis possible. He also taught me how to write academic papers to stand out our strength and it definitely will be of enormous benefit in my future research as well.

Thanks to all of my lab members in 5317 HP. Especially, I would like to thank Ling Xu, an old friend since we were in Saskatoon, who helped me a lot in the multi-faceted aspects of life and whom I shared the moments of failures and success with. Thanks to Raj Wasson who gave me the first training of doing user experiments with patience and shared all of his resources with me generously. I wish next time if he travels to China again I could be the honored guide. Thanks to Abdolreza Shirvani who shared a lot of interesting stories with me and helped me enhance my slides and presentations through his careful scrutiny. I am also grateful to make a friend with Yang Cao for her cool ideas and her witty remarks on the Ph.D life. There are many other lab persons who, in my opinion, have contributed to this endeavor. I'd like to thank Gail Carmichael, Jacquelin Caron, and Jamie Madill. Those candid conversations with graduate life and business ideas improve my thoughts about my own work and spark my other interests. Thanks to all of the participants for taking their time and giving their sincere comments on my experiments.

The members of my supervisory committee (Doron Nussbaum, Jochen Lang, and Jim Davies) are invaluable. Specially, Doron supported me a nice reference generously wherever he could and always suggested me the best choice. Jim showed his selfless insights into my user experiments and helped me sort out the ideas with the studies.

Many thanks to a lot of my friends. I thank Cesar Astudillo for his sharp and smart opinions on life as a parent and research as a researcher. From the beginning, I know he is and will be a very serious researcher. The discussion about his and my research by accident was merged together nicely. I wish to collaborate with him someday and am still crossing my finger to the stippling project we are brainstorming at. I thank Oscar Morales-Ponce, a great friend. The spontaneous friendship was resulted from the conversions on the Ph.D life and academic choices. I thank Chris Hack for those fun when we are teaming for our TA jobs, a nicer man than me. He also generously provided any help he can provide on my experiment study and sincerely gave his advice on improvements to this document through his careful inspection.

Also many thanks to Michael Wiwchar on improvements to this document. Specially, I would like to thank my teacher as well as my friend, Christine Laurendeau. She is one of the best teachers I ever worked with, full of passion and energy, a strict face but with a sincere heart full of care for students. She is an excellent educator model I wish to be.

I am also grateful to have a lot of other true friends in Saskatoon to support me and to make my life colorful: Kelly, Na Bian, Dong Liu, Xinghui Zhao, Xue Li, Dange Huang, Xia Liu, Hengming Zhao, Cythiya Chen, Jun Liu, and Ran Chen. And I also want to thank the professors Mark Eramian and Nadeem Jamali at University of Saskatchewan who gave me a lot of help in life and in research.These acknowledgements will not be complete if I did not include my gratitude to a lot of previous researchers: Jianghao Chang , Benoît Alain, YingeQu, Ovidio Mallo, Domingo Martn, and Mingtian Zhao. They responded my questions quickly, provided their software and codes generously, and also told me a lot of implementation details. I also have to mention the many open source tools and libraries which I take for granted make my life much easier and saved the countless hours I spent at the computer. I am also grateful to use many images shared in Flickr.com in this thesis.

The financial support for my doctoral studies was partially provided by the Carleton University, GRAND, and NSERC, and previously by University of Saskatchewan.

List of Tables

List of Illustrations

List of Appendices

List of Abbreviations

AT	Artistic Tessellation	150
CAH	Contrast-Aware Halftoning	13
CED	Contrast-aware Error Diffusion	49
CPSNR	Contrast PSNR	67
CSS	Content-Sensitive Screening	14
CVD	Centroidal Voronoi Diagram	35
ED	Error Diffusion	46
ETF	Edge Tangent Flow	92
FS	Floyd-Steinberg	46
MSE	Mean Square Error	61
MSSIM	Mean Structural SImilarity Measure	62
NPR	Non-Photorealistic Rendering	1
Prioritized CED	Priority-based and Contrast-aware Error Diffusion	12
PSNR	Peak Signal-to-Noise Ratio	61
RAPSD	Radially Averaged Power Spectrum Density	67
SAED	Structure-Aware Error Diffusion	27
SAH	Structure-Aware Halftoning	27
SPS	Structure-Preserving Stippling	14

List of Symbols

List of Definitions

Chapter 1

Introduction

1.1 Non-photorealistic Rendering

For many years, researchers in computer graphics have focused on using computers to simulate real world scenes. One of the goals in computer graphics is to make computer generated imagery look as realistic as possible. Over the past two decades, a new area that does not share the goal of realism, called non-photorealistic rendering (NPR) [41, 46, 149, 171], has emerged. This new area of research is inspired by art and art history to either simulate traditional media or simplify an image in an illustrative form. This thesis investigates NPR algorithms in image space. Given an image, NPR methods can produce stylized images. Image stylization is a process or a method of converting imagery into a new image, in which the new image looks as though it had been created with a traditional artistic technique, such as painting, stippling, or illustration. The stylized image uses stroke elements or colors that preserve useful features inherited from the original image while omitting the image content that is not essential. The content abstracted from the original image is either simplified with main structures, such as object silhouettes, for a clean and clear representation; or emphasized with thickened or highlighted importance, such as edges, by using large size strokes or strong contrast color assignment. For example, to draw a ball using pencil, the contour of the ball is a feature more essential than the textures that the ball surface has. The ball can be clearly drawn by placing strokes to draw the contour and stylizing the contour with thickened or darkened strokes, with no needs to address the textures on the surface.

Image abstraction [20] is a process of generating stylized images by reducing the content in an image, typically in order to selectively emphasize information which is relevant for a particular purpose. The abstraction process preserves useful and important content that is the essential nature of an object, such as internal structures and size, from an original image for a simplified and/or enhanced representation. For

instance, an illustration used in biology textbooks looks clean and simplified, which is easier for students to understand. One reason is that the illustrated objects are drawn with a minimal number of primitives and colors and without displaying the context of lighting. The effective representation for conveying image content brings NPR algorithms many practical applications in education and engineering. The stylized images also have intrinsic beauty and are attractive. Recently, we have seen artistic images rendered by NPR algorithms in computer games, films, advertisements, and websites.

Since computers cannot generate all the stylized effects by artists and NPR algorithms cannot replace artists either, there are many interactive software tools such as Adobe Illustrator and Corel Painter to assist artists in producing visual effects. Interactive digital tools enrich artists' traditional toolboxes, such as instruments (e.g., brush, pencil, crayon), with convenient operations for sketching objects, controlling the strokes, and managing their works. The various operations could be handy for professionals who have had practice and training. However, those without training but who want to create their own stylized image, may feel completely lost by the great variety of tools available. Sometimes automatic methods or algorithms with very few configurations and parameters are convenient for those without knowledge of complex skills to obtain results quickly. Professional artists with skills of using software tools can benefit from automatic approaches as well. Artists can save many hours by replacing tedious manual work, such as placing strokes in large unimportant regions, with automatic approaches. Useful information automatically measured by algorithms, such as silhouettes and creases detected by edge detection methods, can provide informative suggestions on the location and orientation of strokes.

However, the effort of improving automatic NPR algorithms for high-quality results becomes especially challenging if there is no user intervention in the process. Given an image, an automatic NPR algorithm can stylize or emphasize the necessary content by placing primitives and the original image can be converted into an artistic image with a certain style. Figure 1.1 shows a portrait photo with a large amount of unimportant details, such as background people, colors for clothes, and even buildings. To stylize this photo, we must distinguish the most important objects (the two

persons) from the background. The selection between important and extraneous content can be done quickly and correctly with our eyes. However, computers have no

Figure 1.1: A portrait photo.

means of prioritizing the data. Because of the lack of intelligent choice, the computer-generated stylization process cannot know which two persons to be portrayed and where to place the strokes in order to depict the two persons. The automation of stylization is one of the most difficult problems in NPR research. Interactive systems can produce high-quality stylized image in terms of content preservation, because artists can use interactive operations to select the important objects and to refine final results by placing the strokes manually until the rendered image attains their quality requirement. Selecting the important objects and placing thousands of strokes manually take a long time for artists to accomplish. Alternatively, if we would like to obtain a stylized image quickly, we can render an image automatically by sacrificing stylization quality. This thesis proposes some automatic NPR algorithms in image space that can be used to create a variety of styles in a shorter processing time and higher quality in structure preservation than previous automatic NPR algorithms.

1.2 Problems

Our thesis focuses on one major problem. Given a photo and a set of parameter values, how can an automatic algorithm produce a rendered image resembling a certain effect, such as halftoning, screening, stippling, line art, and mosaics? In addition, we would like to preserve important structural details, such as silhouettes, creases, and textures. An original photo contains numerous pixels with plentiful colors which indicate objects and background scenes clearly under certain lighting and environmental conditions. The abstracted content in stylized results can represent the main facts contained in the original photo by depicting key objects with important silhouettes and creases and sometimes with necessary tones. Locating the structural content from a large number of pixels is not a trivial problem. Visually, a stylized image is the result of a distribution of primitives. Stylization can generate a rendered image that differs from the original image in that pixels have new colors to help enhance the representation of content, for example, by thickening object shapes and increasing color contrast. The result depicted by a collection of primitives should represent the main content in the original image including main objects with correct locations and size information and internal structures of objects; and the skipped content should not affect or change the major knowledge delivered by the input image. Our goal is to create stylized images with improved quality in preserving structural details.

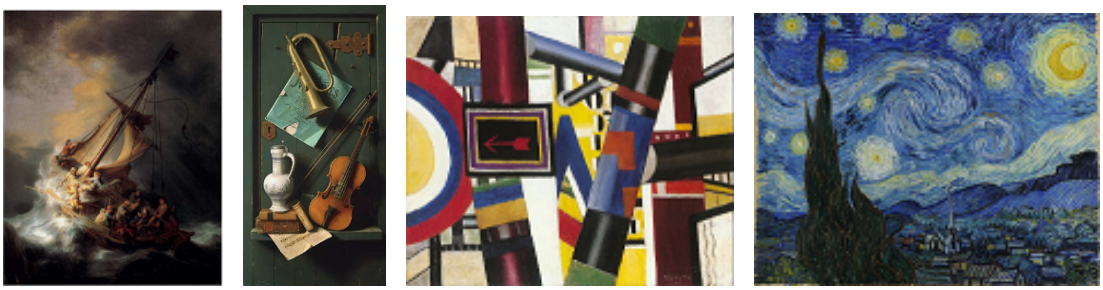
In addition, we address a problem for pattern creation including natural and abstract patterns. Patterns that are formed by the power of nature are called natural patterns, such as bark, cracks, fish scales, and zebra stripes, which contain randomness and irregularities because they are long-term results from influence of climate and physical and chemical interactions with other objects. Abstract patterns are patterns that represent no natural or actual objects in the real world. For example, there are abstract patterns that are created by humans, such as, patterns on pottery and geometric and decorative designs for architecture and textile. The creation of abstract patterns involves with a composition and a process of assigning various elements such as color, line, shape, and form. We investigate the placement of curves to create patterns. Specifically, the additional problem can be described as follows: given a drawing space and a background vector field, how can an automatic placement of curves build a partition for natural or abstract patterns? We address this additional

problem by proposing rules and strategies of placement.

No single algorithm can cover all styles. We pick some typical applications and styles to demonstrate the solutions to the problems posed earlier. The complete combinations of different attributes for different primitives are extremely broad. Therefore, we choose to concentrate on six applications: halftoning (pixels), screening (pixels), stippling (dots), line art (lines), patterns (curves), and stained-glass mosaics (curves). These applications cover a broad range of effects and styles in NPR research and personally we like these styles also.

1.3 Perceptual Motivation

One important source of inspiration for this thesis has come through the observation of artistic works, such as paintings and drawings. When an artist creates art, the placement of strokes depends on the artist's intention [30]. For example, in Figure 1.2 he or she may intend to tell a story (e.g., *Christ in the Storm (1633)* by Rembrandt);



(a) (b) (c) (d)

Figure 1.2: Artists' intentions. (a) A story; *Christ in the Storm* by Rembrandt; (b) realisms; *Old Models* by Harnett; (c) forms; *Railway Crossing* by Lèger ; (d) expression; *Starry Night* by Van Gogh.

or to create a realistic environment (e.g., *Old Models (1892)* by Harnett); or create forms (e.g., *Railway Crossing (1919)* by Fernand Léger); or to capture an impression (e.g., *Starry Night (1889)* by Van Gogh). When we perceive the strokes painted by artists, we not only can acquire knowledge and stories delivered by artists, but also can feel their emotions.

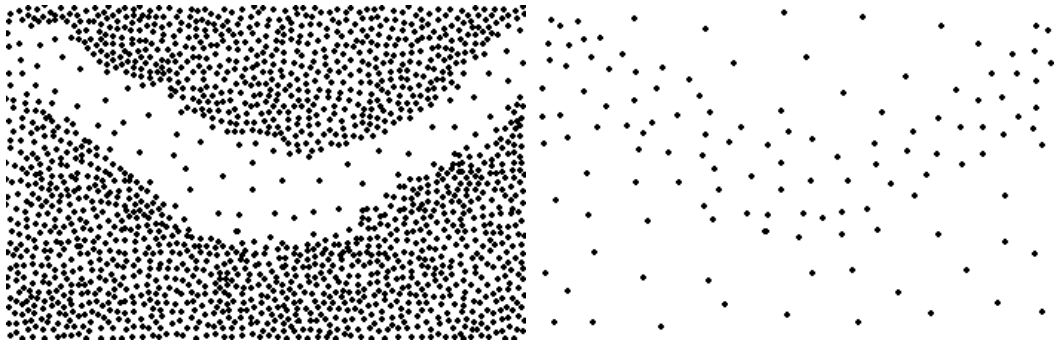
Figure 1.3: *Madame Palmyre with her dog*; Henri de Toulouse-Lautrec 1897.

1.3.1 Importance of Structure Preservation in Stylization

When a viewer perceives an object, there are intrinsic and extrinsic properties he or she can recognize. Intrinsic properties belong to the essential nature of a thing, such as size, shape, and the inner structure of the object; and extrinsic properties are additional details added to enrich the intrinsic properties with colors, shading, and shadows. For example, in Figure 1.3, artist Toulouse-Lautrec did not draw every single hair for the woman. Insead he depicted her hair with very few long strokes by sketching her hair shape and indicating dark and light areas for lighting and hair volume. He also sketched her face silhouette and the outline of her long-sleeved jacket with a few clean and long strokes. Missing such key information about her face and clothes makes the drawing of woman incomplete and vague to viewers. Conversely, the artist ignored the trivially extrinsic properties such as the color and the texture

of her clothes, the background scene, the details of shading in the lighting, and the accuracy of shadows. Even without the additional details, a viewer can tell from the face contour and body shapes that this woman is not young and is smiling. Not surprisingly, in creating artistic works, artists pay attention to the key information that has to be displayed while they deemphasize extrinsic properties of scenes or objects in that omitting the content that are not essential does not impair the main intention, but strengthens the information delivered.

As for our major problem, we anticipate that, when a photo is abstracted for a style, the selected content will represent the key properties of objects in original image including size, shape, and structures. However, the key properties of objects in an image are not available directly. The direct data from the input image contains thousands of pixels. Each pixel in greyscale has an intensity indicating the lightness at the location in image space. An image depends on the change of intensities to depict objects. The gradual change in intensities can convey the tone of a smooth surface (e.g., faces in Figure 1.1). Researchers have focused on matching the tones in stylized images with the tones from the original image. Halftoning research, for instance, has a large amount of work on improving tone matching. Floyd-Steinberg [39] and Ostromoukhov [134] used error diffusion algorithms to generate halftoned images, which solely maintain tone. Similarly, in stippling generation, Secord's stippling method [161] uses tone matching to display the lighting change in an image reference. Screening approaches [174] employ a set of dithered patterns based on tone to approximate a continuous shade in black and white. However, the abstraction quality from tone matching does not satisfy the researchers' desire for high quality in structure preservation. The abrupt change in intensities of the pixels (e.g., hair in Figure 1.1) are edges that indicate silhouettes, discontinuous creases, distinct changes in lighting condition, or textures. Research on image processing and computer vision has proposed many edge detection algorithms [166]. NPR researchers [13, 34, 37, 91] often borrow those edge detectors to obtain the key structures for stylization. NPR researchers recently attempted to include structural awareness into halftoning [15, 97, 139], stippling [84, 100, 123, 165], screening [99, 142], mosaics [27, 98, 104], and other forms of image abstraction [124] so as to retain the



(a) (b)

Figure 1.4: Contrast in dot distributions for representing a V-shape edge. (a) Strong contrast; (b) weak contrast.

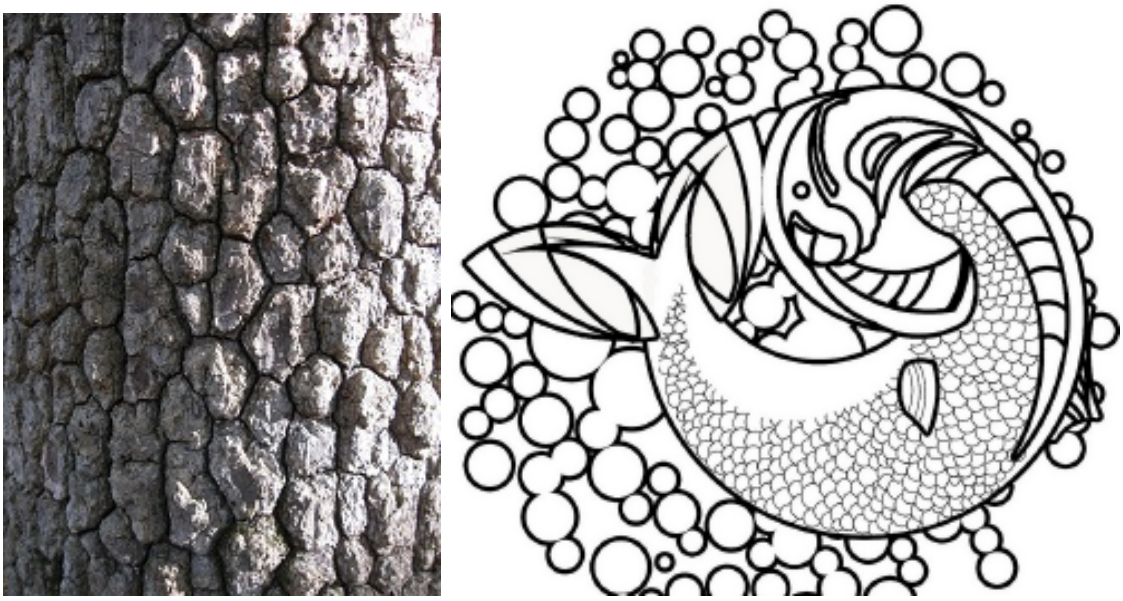
content of the image reference. There is a trend in current NPR research that pursues structural quality. Responding to this trend, the approaches proposed in this thesis aim to improve the matching of content structures between original images and the rendered images. Our structure-aware methods are inspired by perceptual effects, such as contrast and similarity.

1.3.2 Contrast and Similarity

Contrast [31, 40, 62] is the difference in attributes of objects, such as luminance, color, size, and density, which makes an object distinguishable. We observed that to depict structural content, the placement of primitives should pay attention to the surrounding placement. For example, Figure 1.4 (a) and (b) show a V shape represented with the same set of dots, backgrounded by two different distributions: dense placement in (a) and sparse placement in (b). Figure 1.4 (a) shows a high-contrast image due to the significant difference in dot density between the interior (the V object) and exterior (the background) of the V shape. The V shape in Figure 1.4 (a) is very clear. Figure 1.4 (b), however, shows weak contrast and the shape is poorly defined as a result of the reduced difference in dot density between the two spaces. Although both V shapes use the exact same dot distribution, human perceives the dot distribution of the V shape in image (b) as denser than that in image (a). This example demonstrates how people perceive objects differently depending on

the context of the object and its surrounding elements. Contrast created by color difference has been employed in NPR research [18, 150]; however, contrast generated from the difference between the number of primitives with limited colors has been used in creating halftoning and stippled images [15, 161] and is not thoroughly explored in the NPR literature. We propose prioritized contrast-aware error diffusion method, described in Chapter 3, to improve structural preservation through a contrast-aware distribution of primitives. The method is based on the notion that dark pixels should attempt to maintain their darkness and bright pixels should attempt to preserve their lightness during error diffusion. Dark pixels receive more darkness than light pixels do in diffusing negative error and light pixels receive more lightness in diffusing positive error. As a result, the difference in intensities increases. The outcome is that dark areas have more black pixels and light areas have fewer black pixels. The increase of the difference of the number of pixels between dark and light areas can bring a strong contrast image for a clearly defined edge. Assuming that the original photos have clear objects, we can show the difference of intensities by using the difference of the number of pixels, which maintains the contrast. The mechanics to deal with darkness and lightness also follow the tendency in intensities, which preserves the contrast and then the structure as well.

As for the small problem concerning finding natural and abstract patterns in curve placement, we are interested in presenting a partition with spatial control and similarity. In our environment, there are naturally occurring patterns, such as cracks, bark, scales, and erosion formations. Figure 1.5 (a) shows an example of natural patterns of tree bark. Although the bark has different tiles in shapes and textures, each tile looks similar because cracking forms Y shapes that visually are alike. The similarity, irregularities, and randomness of these natural patterns appear beautiful. We hypothesized that one of the reasons for aesthetics of these patterns is the similarity presented in the patterns. Similarity can be defined as the resemblance of two elements in terms of features or qualities [140]. Figure 1.5 (b) illustrates an example using similarity for an artistic design. Notice that in Figure 1.5 (b) that the scales are different and are distributed over the fish surface. Each scale looks similar in shapes but not identical: there are, for instance, variations in size, shape, and rotation. The similar shapes give the image a unified visual appearance that looks harmonious to



(a) Bark (b) Design

Figure 1.5: Natural patterns and an artistic design using similarity. Adapted images from Flickr.com.

us. Similarity here implies repeated, controlled lines to place. Our proposed artistic tessellation technique, described in Chapter 7, employs similarity and also combines the irregularities and randomness in the generation process to create natural and abstract patterns. Patterns are formed by growing curves. We can use similar curves, similar starting direction of curves, and similar growth field to present similarity in curve placement. A small amount of randomness is added to the growth of curves in order to produce irregularities in tiles.

Beyond contrast and similarity, another perceptual effect mentioned a little in this thesis is occlusion. In visual perception, occlusion occurs when an object covers other objects and the occluded objects are either partially hidden or disappear behind the occluder. In an image, occlusion is apparent when two objects' silhouettes intersect on the 2D image plane. We perceive that the object that is occluding is closer than the occluded object. Occlusion indicates depth order. The small fin on the fish body blocks the background scales in Figure 1.5 (b) and suggests that the fin is closer to our eyes than the scales. We demonstrated this occlusion condition by using curve placement. We also provide more details related to perception and art in Appendix A,

including perceptual grouping.

1.3.3 Perceptually-Based NPR Algorithms

A long time ago, NPR researchers had realized the inseparably close connection between art and perception, but this connection has not been fully explored in NPR research. The reason that perceptual factors have not been utilized easily in NPR research is because how a viewer perceives an image is a mental process, depending on subjective and vague decisions. There are many unknown aspects of perception and there are no clear mathematical formulas for computing data about perception. Consequently, the research on perception-motivated NPR algorithms [45, 187, 202, 203, 204] progressed slowly.

Several perception-based approaches to stylization focus on proposing image-based filters to enhance important features by managing contrast in colors. Winnemöller [187] is interested in perceptual importance and proposed filter-based approaches for automatic image abstraction. The increase of local contrast for important regions caters to edge-sensitive cortical cells. His approach polarized contrast in luminance, color, and edges to bring saliency, which made stylized objects in painterly rendering and hatching separate from the background. The stylized images are more easily understood and remembered. Santella [156] employs eye movement recorded from viewers to measure the image importance. His approach can produce stylized images with contrast enhancement. He successfully applied his approach to painterly rendering and colored drawing. Lam et al. [94] investigate brightness perception in cortical area V1, in which cells detect basic features like lines and edges. The stylized images generated by Lam et al. emphasize edges to improve the structural quality. The perceptually-motivated framework they proposed can flexibly and effectively produce various artistic effects including painterly rendering, crayon, and watercolor. Lopez-Moreno et al. [106] approximate depth cue from images. The recovered depth can help stylize input images such as relighting the scene. Although the approximation of depth cue is inaccurate, new stylized images can display different lighting conditions after relighting.

Beyond filter-based methods, many NPR researchers consider perceptual effects in stroke-based rendering methods. Healey [52] investigates the possible issues of

primitive placement related to NPR and visual perception. Later, Healey et al. [53] focus on presenting perceptual salience in stylized visualizations by considering the perceptual features between the properties of strokes (e.g., color, texture, orientation, contrast, and size). They propose a series of perceptual guidelines and build an effective representation of the underlying data. The visualization applications of weather dataset show that their representation is effective. Xu and Kaplan [193] adjust the boundaries from the segmentation in a graph to obtain high contrast between regions, which provide their black and white style high structure preservation. Rosin et al. [150] also propose a stylization method for black and white from the assignment of black and white blocks to strengthen the difference around the edges. They obtain good quality because of contrast consideration. Recently, Zeng et al. [202] and later Zhao et al. [203] successfully employ perceptual attributes (e.g., contrast, saliency, and orientation) and explicitly emphasize contrasts in their placement of strokes during rendering images. Their approaches are able to differentiate primitive properties between dry versus wet, hard versus soft, and long versus short. The control over the difference in stroke attributes brings stunning artistic results in the final paintings.

Our proposed NPR algorithms aim to improve structure preservation. We focus on presenting contrast and similarity in primitive distribution. The exploration in this thesis will be one of only a few automatic stylization systems motivated by perceptual effects.

1.4 Contributions

The thesis thread provides a set of algorithms to create a group of effects: halftoning, screening, stippling, line art, and mosaics. Herein, we propose two novel and general frameworks for a family of NPR styles:

1. The first technique is called Prioritized Contrast-aware Error Diffusion (CED): a technique that uses a flexible priority-based scheme for contrast-aware error diffusion to produce abstracted imagery automatically. Traditional error diffusion methods [35, 39, 134, 177] had superior speed in computation, but were weak in their ability to create high quality images. The halftoned images generated

by conventional error diffusion methods lack structural details and annoying artifacts occur. Our proposed method shares high speed with traditional error diffusion and simultaneously introduces more applications and styles, including screening and stippling, with very high quality results in structure preservation. The basic idea is to maintain the initial tendency of original pixel intensities and thus preserve the contrast of the input image. The new priority scheme breaks the conventional raster scanning order used in traditional error diffusion methods and uses a more flexible processing order. Beyond halftoning, the technique of Prioritized CED can create a set of styles, including screening, stippling, and line art. This thesis demonstrates many high quality stylized images in terms of structure preservation.

2. The second technique is an Artistic Tessellation (AT) scheme based on particle tracing. Our AT method grows curves from a particle system to produce artistic partition automatically. AT is used to present similarity with spatial control in curve distribution to create natural designs. The growth of partitions from curves provides flexible control over tile shapes. In the NPR literature, stroke-based algorithms and tile-based methods have been considered separate. Our AT approach seeks a transition from primitive placement to region distribution and has the ability to control properties of curves, such as curvature. Moreover, the application of this idea in creating cracks, scales, and rivers shows how this exploration can be valuable and productive for natural and abstract patterns in digital production.

1.5 Overview of Results

We propose two frameworks to produce high quality stylizations according to various visual elements: pixel, dot, line, and curve. An overview of the results presented in this thesis is given as follows:

- Contrast-Aware Halftoning (CAH) converts a continuous-tone image into a pattern of black and white pixels with superior structural preservation compared to advanced/current structure-aware halftoning [15, 139] and with competitive

speed as well. Its merits come from its ability to illustrate weak edges, which are normally lost by other halftoning methods [35, 39, 91, 134, 177, 205].

- Content-Sensitive Screening (CSS) concentrates on presenting more varied patterns. Rather than introducing unsatisfactory uniform patterns like traditional dithering methods [174], the CSS results smoothly unify different patterns with clear image content without introducing segmentation artifacts.

- Structure-Preserving Stippling (SPS) uses prioritized contrast-aware error diffusion to create stippled images that preserve structural details with some tone suggestion. The stippling resulted from SPS can express very complicated and detailed content, which is not easy to automatically achieve. The final approach is a fast, automatic structure preserving stippling method that has the ability to produce diverse new styles such as heightening and scratchboard.

- The thesis also presents a new method for creating line art styles. It produces simplified drawing style, an emphasized drawing style, and a combination style with stipples and lines. The simplified drawing style abstracts image content to an extreme degree. Very long lines are employed to express object silhouettes and a few short lines provide tone indication. The emphasized drawing style and the combined style with stipples and long lines enhance key features with large-scale structure.

- Artistic tessellation can create abstract and natural patterns such as cracks, scales, and waves. An organized stack of curves can produce the illusion of a 3D shape. Making 3D illusions by stacking curves is brand new area of NPR research and may open up new directions for future research.

- One application of AT is to generate stained-glass mosaics. When compared with existing mosaic methods [27, 50], the mosaic results maintain good texture indication; thus, making it attractive to produce highly-textured content, which was a weakness of previous mosaic methods.

1.6 Organization of Thesis

The thesis is organized as follows:

- Chapter 2 presents previous work in the NPR literature including media simulation, NPR tools, and stylization. The main focus is on previous work on primitive-based stylization. It also discusses existing research on perception in NPR.

- Chapter 3 describes the technique of priority-based and contrast-based error diffusion, Prioritized CED, in detail. It shows how to use Prioritized CED for halftoning. The chapter compares the quality of contrast-aware halftoning with other halftoning methods [15, 39, 134, 139] in different aspects: tone quality, structure quality, contrast quality, and visual quality.

- Chapter 4 provides two variations on Prioritized CED: exclusion-based masks and a multiple stage priority configuration to create content-sensitive screening.

- Chapter 5 proposes a variation on Prioritized CED for stippling generation. Variations are applied to stippling in order to create a variety of styles.

- Chapter 6 presents line art styles. It addresses the conversion from stippling to line drawing with short lines. It also covers the generation of line drawing with long lines.

- Chapter 7 proposes a new tessellation method by growing curves in a particle system. Splitting and stacking techniques are covered to demonstrate the generation of abstract patterns, natural patterns, and 3D illusions.

- Chapter 8 concludes the thesis by showing its contributions and limitations. We also describe future work.

- Appendix A presents background related to perceptual principles.

1.7 Summary

This chapter provided an introduction to this thesis, which first describes objectives and motivation of the research. It also provides an introduction to the contributions of this thesis and finally, summarizes the results of the thesis. A list of publications based on this thesis follows:

- Hua Li and David Mould. Contrast-aware halftoning. *Computer Graphics Forum*, 29(2):273–280, 2010
- Hua Li and David Mould. Artistic tessellations by growing curves. In *Proceedings of the 6th international symposium on Non-photorealistic animation and rendering*, Non-photorealistic animation and rendering '11, pages 125–134, New York, NY, USA, 2011. ACM
- Hua Li and David Mould. Structure-preserving stippling by priority-based error diffusion. In *Proceedings of Graphics Interface 2011*, Graphics Interface (GI '11), pages 127–134. Canadian Human-Computer Communications Society, 2011
- Hua Li and David Mould. Content-sensitive screening in black and white. In *International Conference on Computer Graphics Theory and Applications (GRAPP)*, pages 166–172. SciTePress, 2011

Chapter 2

Related Work

2.1 Introduction

This chapter gives an overview of the existing research on NPR. It begins with an introduction to NPR, followed by a discussion of various categories of research: media simulation, tool development, and stylization. We put significant attention to individual styles based on primitives. This chapter covers previous work on two pixel-based applications (halftoning and screening), a dot-based style (stippling), a line-based style, and a region-based style (mosaic). This chapter ends by describing previous work on perception in the field of NPR.

2.2 State of the Art of NPR

NPR algorithms have been increasingly and widely applied to our everyday life, as seen in computer games (e.g., NPRQuake, Borderlands), advertisements, and computer-generated films (e.g., Ryan, a Scanner Darkly). Current NPR algorithms can produce stunning synthetic results and animations, which are sometimes similar to the realistic look of traditional artwork. From this point of view, the field of NPR is an interesting and potentially rewarding area of research to pursue. The work in this thesis investigates different styles in practice.

This chapter follows three classifications proposed by Gooch and Gooch [46] to describe NPR research: media simulation, software tools, and artistic stylization. Among these three branches, this chapter pays more attention to previous work in stylization. We focus on image space to address the problem of image abstraction and address artistic partitions in 2D space based on vector fields and initial primitive distribution. All of our results are stored as images or vector graphics.

2.2.1 Media Simulation

The NPR research on media [46, 171], including materials, tools, and substrates, is used to simulate traditional artistic styles created by artists. Different artistic styles employ different substances (e.g., oil, acrylic, wax, watercolor), instruments (e.g., brush, pencil, crayon), and substrates (e.g., canvas, paper, wood). The goal of the simulation is to produce marks that are indistinguishable from the real media seen in art works. In many situations, photorealistic techniques such as physics-based models [141, 143] play an important role in media simulation, but this method of simulation is very slow. Recently, example-based methods [4, 66, 117] with statistical models and texture synthesis have developed a comparable quality to physics-based models with improved performance. Hence, example-based methods attract more and more attention in the field of NPR. Below we discuss three popular media systems: pencil drawing, watercolor, and oil painting.

Pencil, Ink, and Charcoal Drawing

By using different media when drawing, such as pencil, ink, and charcoal, many different visual effects are created in line art. The inherent characteristics of physical materials determines the features of realistic appearance. Sousa and Buchanan [167, 168] proposed an observational model to consider micro-interactions between pencils and paper, as well as the outcome of blenders and erasers. Although Sousa and Buchanan's results simulated the look of hand-drawn pencil drawing, their method needed heavy computation for simulation. Mao et al. [115] and later Yamamoto et al. [195] proposed image-based methods to transform an image into a pencil drawing style by using line integral convolution. The results of image-based processing roughly displayed orientations for pencil strokes. Mao et al. and Yamamoto et al.'s methods were faster than physics-based methods, but with much lower quality. Recently, Lee et al. [96] accelerated pencil rendering for 3D meshes on a GPU. They proposed a shaking technique for sketching contours in order to imitate the effect of trial-and-error of human techniques. They then analyzed the characteristics of pencil drawing by choosing 32 pencil textures and considered the tone variations and texture orientations. By carefully considering the quality and efficiency of algorithms, Lee et al. created impressive real-time animations.

Winkenbach and Salesin [185, 186] did excellent NPR research on pen-and-ink illustrations. Winkenbach and Salesin started with 3D geometric models and described the artistic principles of generating pen-and-ink illustrations: conveying shape information, creating tone and texture, and casting shadows using strokes. Later, other researchers [26, 93, 153, 184] followed the framework of Winkenbach and Salesin to improve the quality of pen-and-ink illustrations. Charcoal drawing [9, 112] was generated in this way also, but focusing on media effects from charcoal. However, automatically creating drawings using lines from image space is rare, and therefore image-based drawing has much room for improvement. Automatic line drawing methods have to find out where to place lines. Edges from edge detectors are good ways to extract the important objects from an image, but image abstraction has further steps, for example, to select the necessary edges and to determine the representation of edges, e.g., the ways of placing strokes. Showing edges is not enough for image abstraction.

Watercolor

Watercolor is an attractive style created by using paper, water, and pigments. The subtle variation of color saturation, in addition to the interesting interactions between the pigments and the paper, clearly distinguishes watercolor from other styles. Curtis et al. [21] provided the fundamental work of watercolor simulation: including effects such as edge-darkening, granulation, backruns, separation of pigments, and glazing. The framework was a three-layer physically-based system, which simulated fluid flow and light interaction. Later, inspired by Curtis et al.'s work, Lum and Ma [111] proposed a multilayer system to maintain frame coherence in animation. The results included a new drying effect and a brush-like texture by line integral convolution. Recently, the introduction of more complicated details, such as dispersion effects [16], demonstrated the ability to present a more realistic look for watercolor effects. Real-time animations for high quality watercolor results with good performance [10, 11, 109, 110] also exist in the NPR literature.

Oil Painting

NPR research on oil painting mainly focuses on presenting an impression of brush strokes, which belong to painterly rendering [51, 202]. One reason for the focus on brush strokes is that the realistic look of the surface of oil paintings has micro- and meso- structures. Shadow and shading caused by those structures play a crucial role in creating the appearance of oil paintings. How to model realistic strokes with shadows and shading is a very hard problem. Some NPR works for creating oil paintings [7, 89, 90] simulated fluidity of the paints by using research on fluid simulation. You et al. [198] adapted smoothed-particle hydrodynamics and simulated viscoelastic movement of pigment. They demonstrated various situations, including diffusion and absorption between pigment and paper. However, a simple way for painterly rendering [120] is to employ texture (or brush) examples from artists to avoid complicated physical simulation. Very recently, Zeng et al. [202] and Zhao and Zhu [203] proposed an interactive system to present oil paintings by using brush examples from professional artists. The painted results look indistinguishable from real oil paintings.

2.2.2 Exploration in Developing Tools

Another major research branch in the field of NPR is to develop software tools. The software allows users to manage artistic works step by step through input devices such as mouse, keyboard, and digital tablet. The use of software also tremendously reduces repetitive tasks by using automatic approaches of placement. For example, artists can automatically fill a space with thousands of strokes. More sophisticated software allows users to have freedom when choosing layouts, designs, primitives, placements, and individual properties for each element. There are many commercial and noncommercial products, such as Adobe Photoshop, CorelDraw, and GIMP, that provide a rich set of tools and functionality. However, in order to produce high quality results, users must not only possess artistic skills and dedication, but also acquire the skills to use those tools.

Since the first interactive graphics system, SKETCHPAD, was presented by Sutherland [172], there are many interactive systems in NPR such as SKETCH [200, 201], Teddy [63], Harold [17], OpenNPAR [48], JOT [68], and RenderBots [158], to provide quick sketching and editing. SKETCH [200, 201] presented a gestural interface

based on simplified line drawings. It was mainly designed as an initial tool for a storyboarding system. The operations were implemented for rapidly conceptualizing and editing 3D scenes. However, there were many flaws in practice, such as the difficulty in obtaining correct gestures in a short time. Teddy [63] was a modeling tool. Teddy allowed users to draw 2D silhouettes and then inflated the regions to 3D round objects. The system was robust and flexible for modeling numerous objects, but Teddy failed to handle unexpected strokes and it was difficult to express a variety of shapes with arbitrary topology. Harold [17] provided a collection of planar strokes to construct a 3D world by reorienting strokes in a view-dependent way in order to display the camera's movement. Users can easily create extremely asymmetrical objects with a large class of selections. Unfortunately, the inaccurate approximation of view calculation which creates a significant problem when reconfiguring the 3D scene in a new point of view. WYSIWYG NPR proposed by Kalnins et al. [70] can provide artists with controls on stroke and paper texture. The WYSIWYG system directly allows users to place 3D strokes on the 3D models. However, the styles are very limited and users have to place each stroke manually, which is tedious in most applications. JOT [68] is a stylized renderer, which not only implemented the idea of WYSIWYG artistic rendering for 3D models, but also implemented coherent stylized silhouettes [69] and suggestive contours [22]. JOT allowed designers, artists and animators directly and flexibly to produce stylized images and videos. OpenNPAR [48] was a conceptual system that attempted to integrate many styles for a variety of users when rendering both 2D and 3D into styles. Potential limitations emerged from the performance and the effectiveness of the individual algorithms used by the system. RenderBots [158] offered a multi agent system for placing strokes for styles such as stippling and hatching. Edgebots, one of the RenderBot types, had a behaviour to place strokes as closely as possible to the edges of an image, which brought the good representation of the structure and the form of objects. RenderBots allowed users to interactively control the image generation by changing the parameters and the number of agents being used. However, the usage of randomness sometimes weakened the process of edge improvement and a good result took a very long time to obtain.

2.2.3 Stylization

Stylization is a method or a process dedicated to converting photographs into artistic images in a particular style, and is a very productive branch in the field of NPR. This subsection covers a general overview of stylization research. Numerous styles exist in the NPR literature and we will discuss some styles and effects that we will explore, including halftoning, screening, stippling, line drawing, and mosaics. Since our main interests are in primitive placement for stylization, we review stroke-based rendering methods. Many stylizations in NPR, such as halftoning and stippling, have been treated as problems of primitive distribution. Given an image, if we want to convert it into a stylized image, there are a few things that we have to think about: what to depict for this image, where to place primitives, and how to draw the primitives.

Hertzmann [56] classified automatic NPR methods for placing strokes into two categorizations: greedy algorithms [55, 185, 186] and optimization algorithms [50, 123, 139]. In general, a greedy algorithm does not produce an optimal placement. The system runs in a single pass to create the placement. Strokes are never altered once they have been placed. In the NPR literature, greedy algorithms emphasize important features, such as edges, more directly than optimal methods. Conversely, NPR optimization algorithms [50, 123, 139] commonly have iterative steps to gradually minimize or maximize an objective function. Objective functions are formulated as target metrics (or energy) of quality heuristics. Trial-and-error methods [47] are typical optimal methods iteratively to obtain an optimal placement. Although it seems any stroke-based problem can be rendered by using objective functions, designing an energy function is not a trivial problem. An energy function cannot precisely describe subjective and objective imperfection, or randomness and vagueness, as commonly seen in art creation.

Voronoi-based algorithms [34, 37, 84] are another typical automatic methods to place strokes. In general, the final results by Voronoi-based methods are a placement of strokes after an iterative process, such as centroidal Voronoi Diagram methods. A centroidal Voronoi-based algorithm [50] starts with an initial distribution of sites. Voronoi diagrams are constructed based on existing sites and the centroid of each tile is found. The algorithm moves the sites to the centroids, which provide a new distribution of sites. The process iteratively moves the sites to centroids until the

moving distance is less than a threshold. Voronoi algorithms aim at an evenly-spaced placement of strokes to have a spatially appealing visual appearance. The outcome of Voronoi-based methods usually is a converged distribution after iteratively adjusting placement several times.

Example-based NPR methods are automatic approaches too and can construct similarity functions on statistical models to match the stylized image with artistic work. Normally, there is a reference style on hand and the goal is to transform an image into a similar style. Example-based NPR methods can effectively produce fantastic hand-drawn outlook for most known styles. Hertzmann et al. [58] presented an early framework of example-based methods for different styles, called image analogies, which has been applied to applications including texture transfer, painterly rendering, and artistic filters. Also based on example methods, Jodoin et al. [66] and Barla et al. [4] investigated hatching by emulating artistic works; and later Kim et al. [86] and Martín et al. [117] used example-based methods to generate hand-drawn stippling styles. While on the one hand it seems that example-based methods unify the stylization problems, on the other hand, the statistical model used in example-based methods cannot cover all quality factors. For example, structural preservation is difficult to achieve in stylized images created by example-based methods.

Beyond placement, stroke-based rendering methods focus on modeling stroke shapes. Each stroke is usually modeled with a few parameters to provide visual variations of results. Sometimes, the results are combined with an enhanced representation for strokes, such as using color differences or using thick strokes. The enhanced representations for strokes improve shapes of objects and assist the figure ground distinction, which helps stylized images display the important content more clearly.

2.3 Stylization based on Primitives

Early work on abstraction relied solely on matching a resulting image with the tone in an image reference because the intensities are easy to compare. The important features of objects in an image, sometime called structural information, including silhouettes, discontinuous, creases, and textures are usually lost in tone-based approaches. However, depicting the structural information in a stylized image matched

with the original content from an image reference can make objects stand out, thus bringing promotions of stylization quality. Very recently, in order to gain high quality final results, some NPR algorithms for creating styles and effects such as halftoning [15, 97, 139], screening [27, 99], stippling [84, 100, 123, 165], and mosaics [104], retained both tone and structural matching. Unfortunately, preserving structural details from a highly-textured image is very hard for NPR algorithms. To retain structural content, an interactive system is useful. Using tools provides users freedom to indicate important image features manually, but manual selections of important content take time. Automatically extracting important features from an image has proven to be very difficult. NPR researchers are confronted with two additional issues in automation. One is where to place primitives and another is how to represent primitives to have clear visual communications. The same placement of sites can be rendered with different choices of stroke attributes to have different visual appearance. This thesis explores five effects, which are halftoning, stippling, screening, line drawing, and mosaics, by using various primitives including pixels, dots, lines, and curves. Previous stroke-based stylization algorithms focused explicitly on finding the important edges. They drew strokes on the edges to emphasize the content from an image. However, explicitly drawing important features such as edges and placing edges on top of stylized images brings unnatural appearance. We propose that, instead of explicitly outlining the structure and the form of objects, indirect description, such as using contrast in an image to guide distribution, may provide a natural look. Our Priority CED technique does not explicitly extract the edge features of an image. We use the original contrast of intensities to guide our system to display the important content through the distribution of primitives. The system's awareness of contrast produces excellent structural preservation, and thus enhances the quality of the stylization. The next section provides a further survey of stylizations based on primitives.

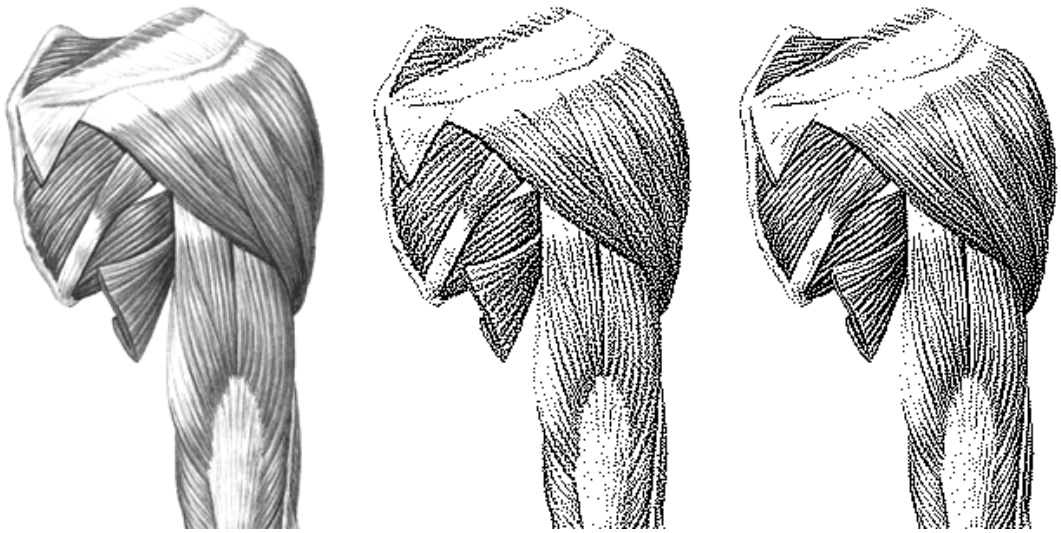
2.3.1 Applications from Pixels: Halftoning

Halftoning is "a process of converting a continuous-tone photograph into a binary pattern in print or on the screen" [95, 174]. Researchers and developers have investigated the subject of halftoning process. Monochrome halftoning is still very

popular and useful for many media, such as newspapers and flyers, with regard to the economical use of ink. Halftoning in black and white is a binary reproduction that uses only one color ink (e.g., black) to print onto another color background (e.g., white). Two examples of monochrome halftoning generated by halftoning algorithms are shown in Figure 2.1. Color halftoning [72, 95] involving many colors is created by generalizing monochrome halftoning algorithms. There are three key factors for the quality of halftoning: the precise preservation of tone, well-preserved structure of the original image, and the absence of spurious patterns. How to satisfy and balance those factors has inspired numerous modern halftoning algorithms, which can usually be classified into three groups: point processes (dithering methods), neighborhood processes (error diffusion), and iterative methods. Dithering methods [8, 174] design a set of patterns for tone levels and replace each location with defined patterns. Error diffusion algorithms are ways to diffuse error to neighboring pixels based on two interests: tone matching [39, 134, 177, 205] and structural preservation [15, 35, 91]. Both point processes and error diffusion approaches run in a single pass. Iterative approaches [1, 3, 139, 192] start with an objective function and repeatedly find the minimum or maximum values of the objective function, which provides the best output of the pixels based on factors in the function.

Early halftoning techniques focused on the goal of tone reproduction. Ordered dithering [8] represented the desired intensity with a defined pattern. However, there is no smooth transition between each pattern. The patterns are designed only for tone matching. There are two major disadvantages to this group of methods: hardly seen weak edges, and noticeable visual artifacts.

The well-known Floyd-Steinberg Error Diffusion (FS ED) technique [39] was developed to circumvent both of the problems of early halftoning techniques. The diffusion of error encourages smoothness between discontinuous transitions to attempt to maintain the quantization error near zero over the whole output. The quality of halftoned images by error diffusion method is still competitive in tone reproduction. Many recent halftoning methods [15, 97, 139] cannot achieve the same tone quality. Inspired by Floyd and Steinberg's idea, a great deal of work [134, 177, 205] has been done in error diffusion. Spurious patterns and lack of structural preservation, however, remain two significant drawbacks for FS ED methods. To improve visual



(a) (b) (c)

Figure 2.1: Halftoning. (a) Original image; (b) structure-aware halftoning [139]; (c) structure-aware error diffusion [15].

quality, a widely-accepted solution for visual artifacts is to enforce blue noise properties in halftoning [3, 88, 121]. Ostromoukhov's halftoning method [134] can generate a distribution with blue noise properties. His method used an off-line minimization process to find corresponding weights for 256 grey levels. Ostromoukhov provided each grey level a different distribution coefficient. Ostromoukhov's method removed much of distracting artifacts and achieved an appealing visual quality for halftoning. Nevertheless, it still did not attempt to retain structure and thus lost fine details of texture. Schmaltz et al. [159] proposed an improved model for halftoning based on electrostatic principles. Their system was the first one to allow points having different color, size, and shape and used an optimized distribution, but preserving the content of an original image is not its goal.

Recently, the quest for high-quality halftoning inspired a trend in structure-aware methods of halftoning. Historically, edge enhancement had been used to promote image structure. Eschbach and Knox [35] adapted the error diffusion method in an image-dependent threshold process in order to increase or decrease edge enhancement. However, Eschbach and Knox did not succeed in rendering weak edges. Kwak et al. [91] proposed an edge enhancement method for halftoning. The quality of results

from edge enhancement still, unfortunately, lost many weak edges. In 2008, Pang et al. [139] proposed Structure-Aware Halftoning (SAH) and introduced an optimization process to iteratively minimize an objective function that includes parameters to balance tonal control and structural adjustment. The key to SAH is the iterative process to minimize an objective function that contains structural factors. SAH took two minutes to obtain an excellent quality for a 512 by 512 image. In 2009, Chang et al. [15] proposed a direct process, Structure-Aware Error Diffusion (SAED). SAED assumed there was only one clear dominant local frequency in the image. Chang et al. obtained error diffusion coefficients through a calibration that grasped the local frequency information. Structural preservation was achieved by adaptive thresholds from a parametrized anisotropic Gabor filter. Both the SAH and SAED algorithms can be treated as state of the art because their structural awareness brings impressive halftoning results, which prompt many researchers to study this traditional printing technique. However, SAH is computationally expensive, and SAED does not achieve the same high level of structure similarity as SAH. Figure 2.1 shows results from the SAH and SAED algorithms.

Although the difference in intensities of original images can provide contrast in color, researchers cannot directly obtain a good representations for important objects from the difference in brightness of an image. Selection and adjustment on the contrast are needed for a good abstraction. Contrast has hence been used rarely in halftoning research. Li et al. [101] proposed an edge-directed error diffusion method that preserved edge saliency and contrast. The process of diffusion stopped when the system found boundary edges. The algorithm's parameters were adaptively modified based on a binary edge image. However, Li et al.'s method suffered from low quality in structure preservation because the error diffusion is similar to the FS ED algorithm, only based on tone. The SAED algorithm [15] adapted contrast as a perceptual parameter in the calibration step. This contrast factor improved the structure quality significantly. However, the SEAD method was complicated and had a lower quality of structural preservation than the SAH method. Our proposed contrast-aware halftoning method [97], described in Chapter 3, employs contrast to generate high quality halftoning. The results from our halftoning method have slightly higher structure quality than the SAH method and our method runs much faster.

Figure 2.2: Screening for Marilyn Monroe art. Left: image reference; right: richness-preserving Manga screening [142].

Previous error diffusion methods usually proceeded in a raster scanning order. Marcu and Abe [116] presented an order of scanning based on intensity values, but without much discussion and with no dynamic updates to the previous order. Marcu and Abe's method can be seen as a precursor to our halftoning strategy. Our proposed halftoning method, Prioritized CED, has a flexible framework of error diffusion by a dynamic priority-based scheme. This priority-based scheme can be adapted to other styles such as screening, stippling, and line art, not commonly available to previous halftoning methods.

2.3.2 Styles from Pixels: Screening

Screening is a pixel-based style. In printing, screening refers to a process of passing ink through a perforated screen (or pattern) over a region. A good screening algorithm can not only represent tone and important features with reduced levels of tone, but also can show different patterns in results. A computer-generated screening example is shown in Figure 2.2 in greyscale. Existing dithering methods [13, 87] or hatching [169, 197] generated uniform screening simply based on image tone with no consideration of the content of an image. Uniform screenings are not aesthetically appealing due to lack of contrast between patterns. Although a wide variety of screens will enrich visual appearance, we have a question: how can different screens automatically help

halftoning while still providing tone and structure preservation? Similar questions have been asked since the emergence of screening [174].

Classic ordered dither algorithms employed a matrix of quantization thresholds, replicated over an entire image [174] and sometimes combined with error diffusion techniques [87]. The variation of rotated dispersed dither [136, 138] reduced artifacts from raster scanning order and also improved tone reproduction. Velho and Gomes [177] used space-filling curves to present aperiodic patterns of clustered dots based on stochastic processes. Both image-independent or image-dependent dithering methods shared a same problem: unsatisfactory uniform patterns.

Subsequently, researchers made efforts to improve unappealing uniform patterns. Buchanan [13] introduced controlled artifacts by organizing the ordering of regions and error propagation directions, although with limited success. Ulichney [175] proposed a recursive tessellation algorithm to generate dither patterns but did not mention how to apply the process to an image. Procedural screening methods proposed by Ostromoukhov and Hersch [135] generated artistic screening elements by introducing various shapes. Smooth transitions were a problem due to a lack of gradual change between shapes, which they avoided in later work by adding more tones [133, 137]. A more powerful way for generating different patterns is through image-based dither screens [178, 180], which transform a texture to a dither matrix and provides control over the texture. Yano and Yamaguchi [197] used a similar technique. However, all the additions of screening methods are based on enhancing the pattern quality through tone. Screenings with the absence of structural preservation, however, are less attractive than screenings with clear preservation of image content.

In order to maintain image content in screening, either automatic sharpening [13, 177] or manual location of important content by users [169] must be employed. Additional information from 3D models also can enhance final results [170, 179]. This thesis, however, concentrates on image space and not 3D worlds. Recently, structure-aware screening [142] designed for manga styles proposed the notion of color to pattern, which connected the image content through patterns. An example of manga screening is shown in Figure 2.2. Since manga screening used greyscales and not binary tones, its problem is not the same as ours.

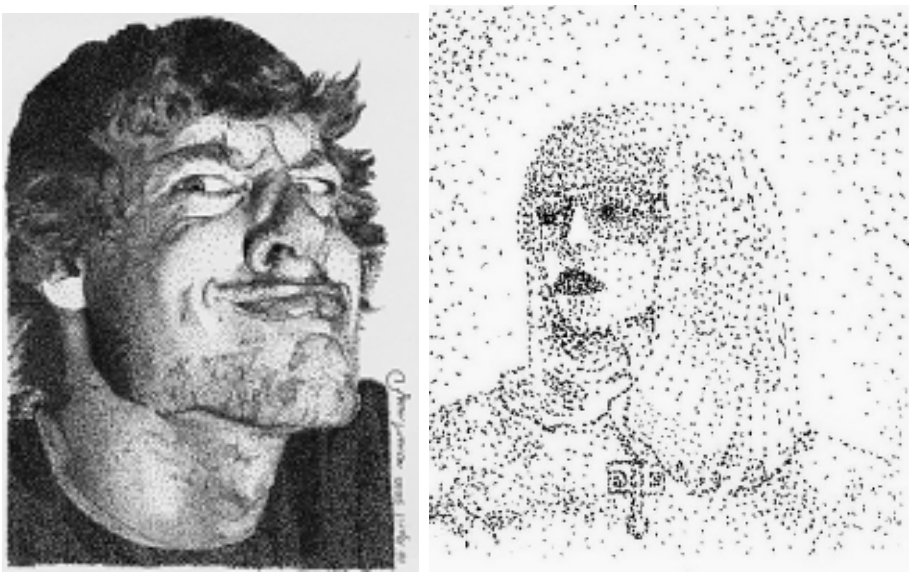


Figure 2.3: Artists' stippled work. Left: a typical halftoning effect by the Flickr user cdslug; right: a stipple effect by the Flickr user makedonche19.

2.3.3 Stippling

Stippling means drawing, engraving, or painting small dots or short strokes onto a canvas or background. A computational stippling algorithm typically seeks a distribution of dots to express the image content. We have seen stipple distributions based on tones for a halftoning effects as well as based on structural content for illustrations. The stippled man shown in Figure 2.3 is a typical halftoning effect. However, people would not say the stippled woman on the right side is halftoning because the number of visual elements is too low to display continuous tone. In early work, researchers were interested in matching tone. Deussen et al. [24] wrote that "a smaller number of relatively large dots is used which vary in size and sometimes in shape" characterized stippling in opposition to halftoning. As the stipple count decreases, the importance of structure concerns will rise, as illustrated by the stippled woman in Figure 2.3. The stipple distribution no longer preserves the tone and loss of tone is inevitable; in order to convey the meaning of an image, stipples must be deployed to structural aspects. We seek to display important features, such as silhouettes of faces, with stipples. Automatically generating stippling with structural awareness [84, 118, 123, 165] exists, but is not common. The structure-guided stippling presented by Mould [123]

barely matched the tone and the edges preserved in Martín et al.'s method [118] are weakened by their emphasis on tone matching. Structural preservation is necessary to retain fine details in the output images. In the very recent NPR book, Deussen and Isenberg [25] distinguished between halftoning and stippling, stating that halftoning is a printing technique and stippling is an illustrative technique.

Iterative relaxation (Lloyd's method) approaches were the early stippling methods employed to create evenly distributed dots [24]. Secord [161] presented gentle tone imitation by weighted Voronoi stippling. Later research on relaxation methods [158, 176] also sought an even distribution of dots. The advantage of relaxation is good tone quality due to even-spaced dots. A distribution with blue noise properties has been considered the ideal stipple placement [24, 61, 88, 158], which reflects a target tone level with even spacing. Kopf et al. [88] employed Wang tiles to ensure blue noise. However, artists often presented stippling with rather irregular appearance, which has a more natural look. Kim et al. [86] and Martín et al. [117] obtained an irregular distribution of stipples by mimicking statistical properties of artists' examples. .

Structural consistency was ignored by many stippling methods [24, 88, 161] that focused on tone matching. Stippling approaches by Kim et al. [84] and Son et al. [165] found objects by segmenting important regions or detecting edge lines, sometimes with user assistance, and stipples were then distributed along these features. The goal of our proposed stippling algorithm is an automatic method. Our approach does not impose much demand on users and favors structural preservation. Some previous stippling methods considered structural preservation. Schlechtweg et al. [158] proposed RenderBots using multi-agent systems to place dots that followed image edges. Vanderhaeghe et al. [176] constrained stroke placement and represented important areas of image content. Kim et al. [84] proposed a method that automatically aligned dots to edges. The concept of directional stippling proposed by Kim et al. was enhanced later by Son et al. [165]. Son et al. presented a better visual appearance of stippling by using a structure grid that enhanced stippled results placed along both the tangential and normal directions, providing illustration-like effects. As we mentioned above, most of these were adapted from iterative relaxation by varying in different constraints. Mould [123] presented a different idea for the structure-guided stippling, which employed a weighted graph and minimized the cost function by path

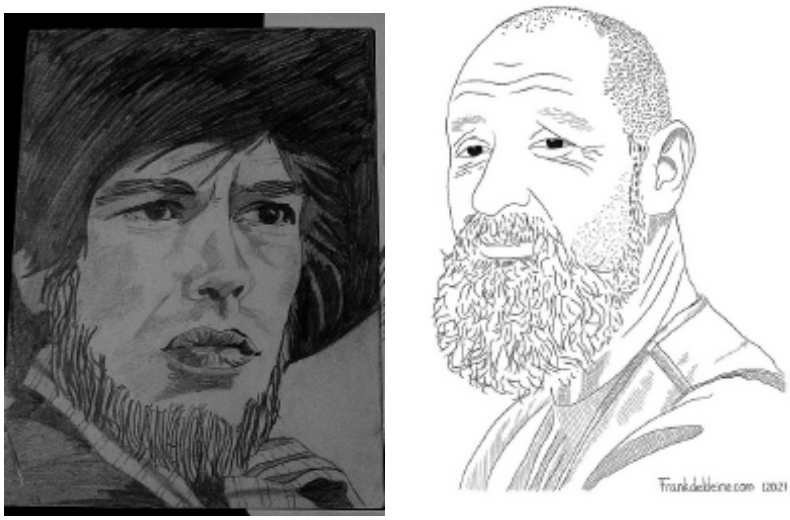
search to place dots. However, he deliberately paid little attention to tone.

In previous stippling methods, it was rare to seek to introduce new styles. Historically, stippling was viewed as its own style and with success measured by tone reproduction [24, 88, 161]. Kim et al. [86] and Martín et al. [117] approximated hand-drawn results by example-based stippling methods. However, their stipplings usually favored faithful tone representation. Kim et al. [84] presented a method to create the hedcut stippling that was based on even spacing along feature edges. Pointillism occurred in the NPR literature too. Jang and Hong [65] and Yang and Yang [196] transformed stippling to pointillism. Other previous tone-based stippling methods [137, 157, 182] addressed the creation of halftoning and artistic styles. The focus on tone created images with lower quality and poor structural preservation. Streit and Buchanan [169] started with importance-driven halftoning and proposed techniques for hatching styles in stippling. In the view of computer graphics, stippling is also a sampling technique. If we can use fewer stipples to sample, an object with a good quality, the reduction of primitives may reduce the calculation after sampling, such as using less calculation for shadows and rendering. With the light computation time, there should be many applications such as in rendering algorithms. The structure-preserving stippling proposed in Chapter 5 extends work on contrast-aware halftoning by adding density control over distribution. The stippling method lends itself to applications in other artistic styles, including heightening, pointillism, scratchboard, and line drawing.

2.3.4 Line Art

A line element has properties of length and orientation. A single dot cannot express directional effects but a line can. The flexibility of a line allows it more easily to represent structure and content than a single dot. Existing line drawing methods focus on presenting hand-drawn effects by simulating media, such as pencil, and the paper texture. However, an image only provides intensities of pixels, without direct data for sketching content. The simplification of an image with very long and very few strokes is very difficult to achieve.

There are many kinds of line art. Two kinds of line drawings are illustrated in Figure 2.4. Figure 2.4 (a) uses many small strokes to fill space, suggesting that the



(a) Pencil drawing (b) Simplified line art

Figure 2.4: Different styles of line art.

number of lines used in a work does not matter as long as it expresses the content clearly. Some lines describe the silhouettes of objects by aligning along the object boundaries, while the remaining lines convey shading. However, some line art pursues a clean look with very few strokes, as shown in Figure 2.4 (b). The long strokes are used to display the large-scale structure of the image, including the face outline, ear shape, and creases of clothing. Some small and short strokes indicate the tone, such as the eyes in the figure. No line is wasted.

Some researchers are interested in presenting drawings with short strokes, which have been seen often in NPR work, such as pen-and-ink illustrations [26, 185, 186] and charcoal drawing [9, 112]. Recently, Winnemöller et al. [188] extended DoG (difference-of-Gaussians) filter to generate high-quality hatching in image space. Their hatchings were very close to the effect of pencil drawings with many small strokes. Secord [161] and Martín et al. [117] extended stippling to line drawing by first creating stipple distributions, and then replacing stipples with lines or strokes. Similar to their extension, we demonstrate drawings with short lines by replacing stipples with lines; however, this direct extension from stippling yields incoherent and broken lines, and produces an effect that is different from the techniques of artists who often employ long continuous strokes.

A clean line drawing with very few and long strokes can clearly display the large-scale structure of an image, illustrated in Figure 2.4 (b). This example pays little attention to tone and only sketches the silhouettes of the man and his wrinkle lines. Despite that a small number of places, such as his eyes and his hair, reveal the tone difference, it illustrates the necessity of tone. Hatching have similar appearance with clean and long strokes as this drawing. Early work related to this simplified style was proposed by Elber and Cohen [33]. They studied how to remove hidden surfaces and extract curves on geometrical surfaces in 3D space. Zander et al. [199] proposed hatching generation, which depicts the silhouettes and hatching patterns with very few and long strokes. Their hatching effect looks very appealing in appearance. Hertzmann and Zorin [60] presented illustrations for smooth surfaces. However, all those method are based on curvature from a polygonal mesh, which is out of our scope. Some researchers investigated interactive tools for creating pen-and-ink illustrations and hatchings. The work done by Salisbury et al. [152, 153, 154] belonged to this exploration. We are interested in presenting an automatic method for this simplified style. Example-based methods used real artistic works as references to generate hatching automatically. Jodoin et al. [66] and Barla et al. [4] investigated this and presented artistic look of hatching. Although Kang et al. [71] and Son et al. [164] have investigated this style of simplified line drawing in image space, their work has provided little advantage over edge detection algorithms in image processing. They applied most emphasis on strong features, and indicated little knowledge related to tone. A very recent paper by Lu et al. [107] investigated pencil drawings. Their work was very impressive, but unlike ours, it used textures and image tone to convey image content, rather than discrete primitives. Our exploration of simplified line drawings can not only express the large-scale structure of an image with very long and few strokes, but can also indicate necessary tone.

2.3.5 Mosaics

Traditional mosaics normally employed small regular tiles such as squares or hexagons to fill in a region. Figure 2.5 shows an example of a traditional mosaic. The outline of the fish is expressed by aligning a set of square tiles along the edges. There is a long history of research on creating traditional mosaics by regular tessellations using

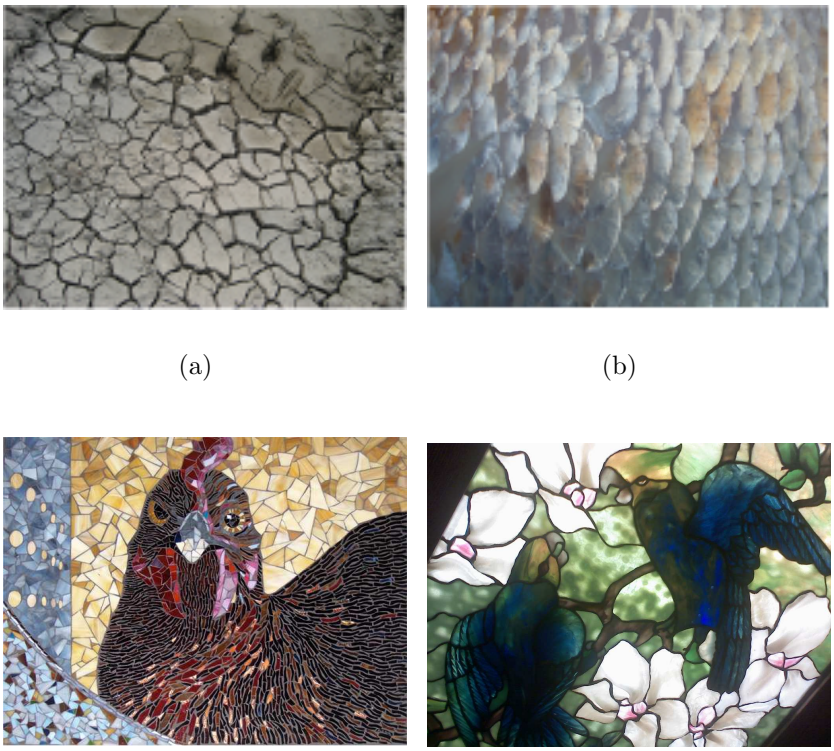
Figure 2.5: Traditional mosaics with regular tiles.

squares or hexagons.

The mainstream of mosaic-making methods is based on regions, typically based on Voronoi diagrams [131], to manage the site placement. Some computer-generated mosaic methods improved the basic Voronoi-based mosaic method by aligning the tiles along the edges to maintain the structure of the image. However, the shape constraints of the use of squares and regular tiles wash out highly textured areas in mosaic results. Hausner [50] was the first person who simulated traditional mosaics using Centroidal Voronoi Diagrams (CVDs). Later researchers [34, 37] followed Hausner's idea and used CVD-based techniques to pack similarly shaped tiles in order to express image content. Smith et al. [163] provided an improvement for mosaic methods by having a distribution following edges from CVD and aligning squares or hexagons with the edges too. In this way, objects in an image can be represented nicely. Recently, Liu et al. [103, 104] used a global energy optimization framework with graph cuts that aimed to solve the mosaic problem. The energy function considered the edge emphasis to improve the structural content. Most mosaic methods used square or hexagonal tiles. We are not interested in pursuing traditional mosaics that use regular or near-regular tiles, but in mosaics with irregular tiles. Less regular tiles existed in mosaic research [6, 27, 132], allowing square tiles to overlap and then cutting the overlap into non-regular regions. However, the varieties of irregular tiles are limited

because of the use of square tiles initially. To introduce tiles of arbitrary shape into mosaics, some researchers treated mosaic generation as a packing problem. Kim and Pellacini [85] divided objects into a collection of containers and packed different shapes into containers. The packing attempted to minimize the space between different shapes. Unlike Voronoi-based methods, Mould [122] generated stained-glass effect with irregular regions by applying operators from mathematical morphology. Later Brooks [12] used a unique region-merging tool to generate stained glass, which formed each tile of mosaics by merging small segmentation shapes to a large one interactively. However, neither Mould [122] nor Brooks [12] attempted to address highly textured structures in regions that contained, for example, hair or feathers.

Irregular patterns often occur both in nature and in artwork. Figure 2.6 shows two natural patterns: cracks and scales. No two tiles are exactly the same; they are random and irregular. Some artists would like to introduce irregular and similar



(a) (b) (c) (d)

Figure 2.6: Irregular tessellation occurs in both nature and artistic works. (a) Natural cracks; (b) scales; (c) Stained-glass mosaic by Barbara Keith; (d) Tiffany glass.

elements into their work. Typical artistic work can be found in Tiffany glass or in stained-glass mosaics. In Figure 2.6, we show an example of Tiffany glass that uses irregular tiles to depict objects. When artists compose real stained-glass mosaics for highly-structured objects with tiles, they usually prefer to represent the large-scale flow of tiles for such textures. The curved boundaries form different shapes that are not as regular as squares. The artist Barbara Keith employed many uneven tiles of varying shapes to express the hen. In particular, the flow of the feathers is illustrated beautifully by elongated tiles. Voronoi-based methods are not commonly used to produce partitions with elongated, irregular, or curved tiles, because those methods manage site locations to change tile shapes. However, the control over site placement is insufficient to generate the desired tile boundaries. Additionally, automatic tessellation methods have difficulty generating a partition that not only includes irregularities, but also possesses texture indication. Most mosaic methods consider aligning with edges in terms of position and orientation and pay attention to object boundaries; however, texture details are typically lost. Recently, Kyprianidis and Kang [92] preserved directional image textures by using directional shock filtering in abstraction; Mould [124] introduced cumulative range geodesic filtering to abstract an image with good texture indication. Beyond the use of irregular tiles, we address texture indication in our mosaics too. Stained-glass mosaics, described in Chapter 7, not only employ different tile shapes, but also demonstrate the beauty of the flow of tiles in highly-textured areas, which is often lost in earlier mosaic methods.

2.4 NPR Research on Artistic Designs

Professional software, such as Adobe Illustrator, provides users with a platform to create artistic designs. We have seen many of these kinds of patterns from our clothing patterns and decorations, to shoes or jewellery. Lu and Steinhardt [108] reported that by 1200 C.E. artists had connected ideas about tessellations and tiling with pattern generation. Early work in NPR research that sought to automatically and efficiently generate good artistic designs, such as decorations, focused on tiling and patterns, e.g. the work by Rangel-Mondragon and Abas [144], or ornamentation designs [191].

Wong et al. [191] enumerated three key factors in ornamentation design: repetition, balance, and conformance to geometric constraints. They employed L-systems

to create floral decorations. Glassner [42, 43] investigated tilings by placing a collection of 2D shapes on the plane with the connection rules to make the patterns having different themes and variations. Later, Kaplan [73, 74, 75, 76, 77] had a series of publications about generating Islamic star patterns, Escher art, ornamental designs, and tilings by using geometrical symmetry and group theory. He also investigated the metamorphosis from one pattern to another pattern. Kaplan and Salesin [79, 80, 81] presented a series of work for tilings and patterns such as Escherization and Islamic patterns. Djibril et al. [28] created Islamic geometric patterns using repetitive tiles based on discrete symmetry groups. The symmetry was also investigated by Kaplan and Hart [78] to create symmetrohedra with attractive appearance. Anderson et al. [2] created hand-drawn ornamentations, such as floral and geometric patterns. The symmetry idea has also been used for generating Celtic knots, such as by Glassner [44], Kaplan and Cohen [82], and Kaplan et al. [83]. Recently, Liu et al. [102] reviewed symmetry-related ideas which had been used in science, computer vision, and computer graphics and they concluded that "symmetry has made a profound and lasting impact".

We are interested in creating an artistic partition by using curves, as illustrated in Figure 2.7. Figure 2.7 presents a drawing example of a cloud using several groups of curves. The curves are stacked in an organized structure over space and the stacked curves resemble the surface of clouds with similar tendencies in curvature. Figure 2.7 demonstrates a 3D illusion from stacking 2D curves.

Our proposed artistic tessellation approach is inspired by the observation of tessellations and nature. We found that each tile boundary in partition is a group of enclosed curves, which can be obtained by a few curves intersected with each other. We also found that, in nature, natural patterns even with similar tiles are random and contain numerous irregularities. We propose to grow curves for artistic tessellations by presenting similarities in curve generation, such as similar growth direction. The results of our approach can express natural patterns such as cracks and scales. Our approach also can be applied to display 3D illusions by growing 2D curves, as demonstrated in Chapter 7. This 3D illusion from 2D curves has barely been explored in previous composition methods.

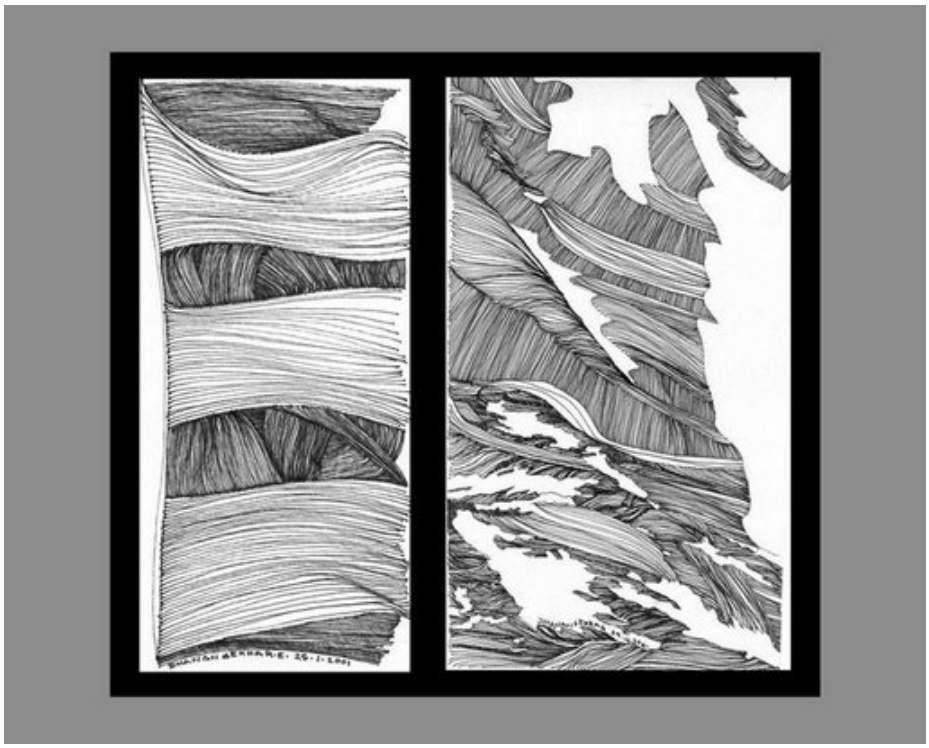


Figure 2.7: Cloud design by the artist Dhanan Sekhar from Flickr.com.

2.5 Perceptually-based Research in NPR

Artists heavily rely on feedback from human visual system. Knowing the connection between perception and art can help NPR researchers to produce stylized images. Perception-based research in NPR can mainly be classified into two categories: perception in stylization and perceptual evaluation. Perception in stylization focuses on how to incorporate perceptual principles into NPR algorithms in order to improve the stylization quality. Perceptual evaluation studies user responses to NPR results through user experiments, aiming to evaluate the quality of NPR algorithms.

2.5.1 Perceptually-Based Stylization

In computer vision [166], there has been significant progress in addressing perception in algorithms. Perception-based experiences in computer vision provide informative clues to NPR studies about how to adapt perceptual principles for algorithms. Historically, automatic abstraction methods [13, 35, 71, 164, 177] employed sharpening or image enhancement methods as a filter or a set of filters to present important aspects

of stylized images. Users can understand the enhanced and abstracted images better because increasing contrast in colors makes the subject salient. Winnemöller [187] further proposed a novel image-based abstraction framework by simplifying and enhancing images to make them easier to understand and remember. His central idea used a non-linear diffusion kernel to increase the effectiveness of non-realistic images for visually-driven tasks, e.g., presenting clearly defined edges. His resulting abstracted images are quite good; however, his exploration was only focused on color adjustment and paid no attention to primitive placement. Consequently, his stylized images, such as his black and white style, are not as good as the previous work by Mould and Grant [125] in terms of structure preservation. Santella [156] assumed that human attention to important areas of an image can guide eye movement. They recorded viewers' eye movement. The system created stylized abstraction guided by the eye-tracking data. demonstrated that the management of the perceptual features can predict the influence on the viewers. The guide of eye movement enhanced important features and made objects look more prominent than previous results. However, recording data from viewers and applying data to stylization are complicated. The data worked only for specific testing examples. Lam et al. [94] overviewed the primary visual cortex and proposed a model of brightness perception using the multi-scale line/edge representation. Ultimately, the artistic results looked not good because the structural quality is strongly based on the quality of the system's edge detection. Their effort exactly showed that explicitly extracting edges and overlaying them on top of final results would not be satisfactory. Lopez-Moreno et al. [106] stylized photos by recovering depth cue from input images. The recovered depth assisted to relight the scene with various interesting styles. The depth approximation, however, was imprecise and occasionally unable to correctly relight an image.

Filtering an image to adjust perceptual effects has limitations on presenting various styles since there are no strokes used. There are many NPR researchers who considered perceptual effects in stroke-based placement. In 2004, Healey et al. [53] learned techniques from artistic works and adapted the perceptual features to adjust the strokes in order to create NPR visualizations with perceptual salience. They used color and texture patterns for perceptual data display and successfully showed Impressionism of paintings for weather dataset. DeCarlo and Rusinkiewicz [23] used

color contrast to highlight suggestive contours to convey shape cues. Very few highlighted strokes were needed to demonstrate shiny surfaces. However, this stylization was presented for 3D shapes, not the image space that we are interested in. Xu and Kaplan [193] proposed artistic thresholds to present contrast between boundaries in black and white by using a graph-based optimization method. However, the boundaries used in artistic thresholding were built upon segmentation methods and the quality of the black and white stylization was highly dependent on the quality of segmentation. Rosin et al. [150] considered object saliency by manipulating the contrast between black and white blocks. The quality of their resulting black and white art was higher than Xu and Kaplan [193]. Both the contrast employed by Xu and Kaplan [193] and Rosin et al. [150] is perceptually-motivated. More recently, Held et al. [54] used a blurring effect to alter the perception of distance and size in virtual environments. Zeng et al. [202] asked users to manually create the semantic graph for an image and incorporated the perceptual characteristics in painting strokes. They can also adjust the contrast by differentiating properties between dry versus wet, hard versus soft, and long versus short. The semantic information is constructed by the manual selection of object boundaries from a user. A later extension by Zhao et al. [203] adjusted the level of perceptual ambiguity for appropriate arousal level shifts, but still the semantic information was acquired through user interaction. The systems proposed by Zeng et al. and Zhao et al. can produce very impressive results because their brush dictionary stored numerous realistic marks from artists, and the system preserved the appealing results of their technique. Both systems by Zeng et al. and Zhao et al. took a very long time to finish the stylized images and the visual effects highly depended on the example strokes that were sampled from the artists.

As we mentioned above, each of those explorations in perception-based stylizations were applicable to a single specific style. Automatic methods for image-based stylization without user selection is hard to obtain. The use of perceptual contrast in NPR algorithms has been investigated by some researchers. However, much further investigation on high-quality results is needed. Our Priority CED and its variations that maintain contrast in placement can automatically produce a variety of styles and effects including halftoning, screening, stippling, and line art.

2.5.2 Perceptual Evaluation in NPR

NPR has developed into a mature field over the last two to three decades. Historically, comparisons of performance between different NPR algorithms used processing speed, which is common in computer graphics. Running time is ill suited to answer questions like: "Does the system provide critical effects for this style?"; "Do different styles have differences in perceptual features?". Side-by-side comparison between images from different algorithms became a common way to evaluate the effectiveness of an NPR approach. However, as NPR algorithms have become more mature, subtle differences in quality are impossible to detect by this raw comparison. Computer vision has objective metrics of the quality of filtered images in terms of tone similarity [166], structural similarity [181], and visual appearance [174], which will be covered in Chapter 3. Those objective measurements have been used directly in NPR to evaluate stylized images (e.g., halftoning), even though it is unknown whether the metrics fit the analysis or not. To the best of our knowledge, there does not exist a metrics of quality of stylization to date.

Hertzmann [57] discussed evaluating human aesthetics, which provided many connections between perceptual evaluation and NPR research. Perceptual evaluation normally means doing experimental studies based on psychophysical research. This style of evaluation has been used often in the field of human computer interaction and the field of psychology. In the recent NPR book, Isenberg [64] reviewed perceptual evaluations. He classified evaluation methodologies [14] into quantitative and qualitative evaluation. Quantitative evaluation focuses on testing hypotheses by using measurable variables in experiments and analyzing the data statistically [113]; whereas qualitative evaluation [113, 126, 155, 160] is focused on employing observation and interviewing in order to acquire opinions from users. Both quantitative and qualitative evaluations have been used in NPR research.

2.6 Summary

The field of NPR is maturing, and the three branches including media simulation, tool development, and artistic stylization are still active. The automation of image abstraction cannot produce the same quality as that of interactive systems used by

artists. Perception has a strong connection, but there is much that remains to be investigated. Very few perceptual characteristics have been applied to the NPR field, but the results from perception-based stylizations are impressive.

Chapter 3

Contrast-Aware Halftoning

3.1 Introduction

Halftoning is a printing technique to produce an output which approximates a continuous-tone image with a reduced number of available tone values. We are interested in generating monochrome halftoning in image space. Our halftoning result is a binary image in black and white, converted from the original image which had 256 graylevels. There are a number of tone values, missing in the halftoning. The difficulty in generating halftoning comes from how to use black and white to display the effects given by the missing tone. The concentration in current halftoning research is not only on good tone matching, but also on high structural preservation. In addition, researchers pursue halftoning results lacking unappealing visual patterns and halftoning methods with a high speed. Structure-aware halftoning [139] and structure-aware error diffusion [15] demonstrated this trend. This chapter proposes the contrast-aware error diffusion framework which processes pixels in a priority-based order to obtain good structure preservation in halftoning.

The idea behind our proposed contrast-aware halftoning (CAH) is simple. Human attention can be attracted by contrast [128, 190]. A monochrome halftoning result is a binary image, which contains only black and white pixels. The difference of primitive density among the distribution displays the contrast. A strong contrast image can clearly define objects, while a weak contrast image displays objects in a vague way. For halftoning, one way to promote contrast is to increase the difference of densities of black pixels between two sides of an edge. The high difference in density can display an edge clearly. Figure 3.1 shows a small part of an artist's stippled work, which demonstrates how to present contrast for a man's face. Since stippled images use only black dots, contrast is derived from the density of black dots. In Figure 3.1, the boundary between the person's hair and the background is clear because of a high difference in dot density on two sides of the edge. The distinction

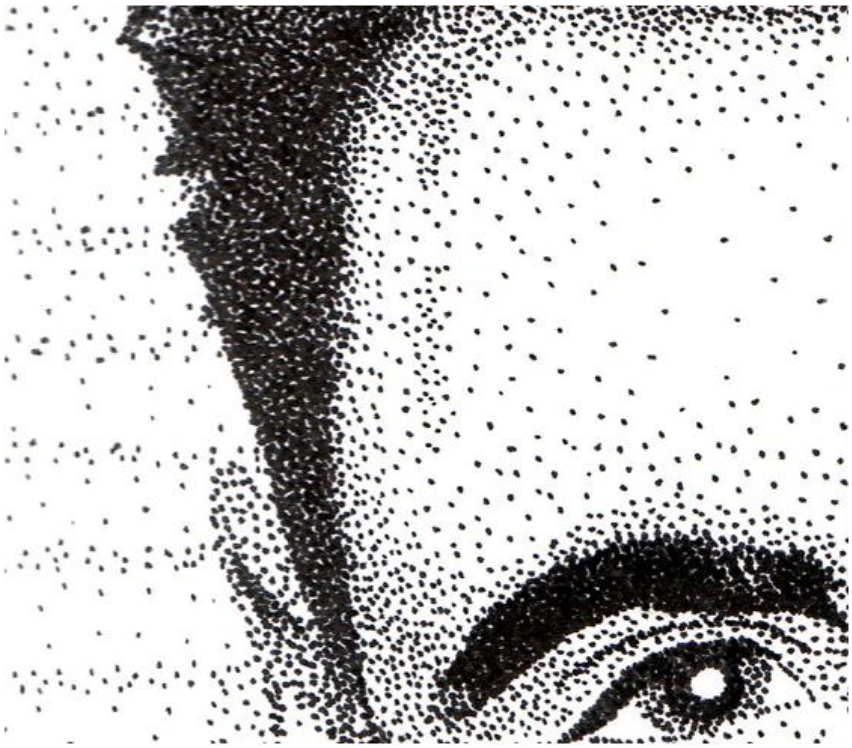


Figure 3.1: Observation of contrast.

between the person's ear and the background is not well-defined because of a low difference in dot density. If contrast at edges is strong, a viewer observes a structure well-preserved. The strategy to obtain clearly defined contents in dot density is to show high differences between two sides of edges.

Contributions. Our proposed basic halftoning method introduces new contrast-aware weight calculation for error diffusion to generate structure-preserving halftoning. We also strengthen this basic halftoning algorithm with dynamic priority. The core idea is to distribute error with contrast-aware weights. The error from a dark pixel is positive, which will lighten the pixels in a mask, and the error from a bright pixel is negative, which will darken the pixels in a mask. The new strategy in weight distribution is to preferentially lighten bright pixels and darken dark pixels. Compared with classic error diffusion, our new error diffusion can bring higher difference in intensities. Bright pixels become brighter in lightening and dark pixels become darker in darkening. Each pixel retains its initial tendency. Error diffusion with

our new coefficients makes dark pixels less lightened, more likely to become black pixels, and bright pixels less darkened, more likely to become white pixels. If the original difference is 10, the modified difference becomes 50. For weak edges, they may not be displayed under the original difference. But they will be likely to be displayed under the modified difference. Dark regions will produce more black pixels and bright regions will create more white pixels and fewer black pixels than classic error diffusion. The increase of difference in the number of black pixels between dark regions and bright regions can bring high contrast halftoning results. Our evaluation of CAH using side-by-side comparisons and objective measurements shows an significant improvement in terms of structure matching. Meanwhile, our halftoning technique generally is faster than structure-aware halftoning [139] and structure-aware error diffusion [15]. We also devise a scheme using dynamic priority to expand stylization capabilities. This ability will be demonstrated in effects and styles including screening, stippling, and line art, presented in Chapters 4, 5, and 6.

3.2 Conventional Error Diffusion

Our proposed method belongs to the class of Error Diffusion (ED) methods. We will give some technical details here for traditional error diffusion approaches. Given an image I in greyscale (an 8-bit image, $I(x, y) \in [0, 255]$), an error diffusion algorithm outputs an array of black pixels and white pixels. The background color in this thesis is white (intensity value = 255) and the foreground color is black (intensity value = 0). For each pixel, the intensity difference between the original intensity and the approximated intensity (output color) – the error (e_{xy})– is distributed to the neighboring pixels in a mask.

The well-known traditional Floyd-Steinberg Error Diffusion (FS ED) [39] is a classic error diffusion method. The algorithm's pseudocode is listed in Procedure 1.

The Floyd-Steinberg error diffusion algorithm scans the image from left to right, top to bottom, thresholding pixel values one by one and outputting black pixels and white pixels. The scanning order is called raster scanning order. The function find_closest_color(*oldpixel*) in Procedure 1 is a simple rounding or thresholding. When the *newpixel* is 0, the system outputs a black pixel; when the *newpixel* is 255, the system outputs a white pixel. The difference between intensities of an old pixel and

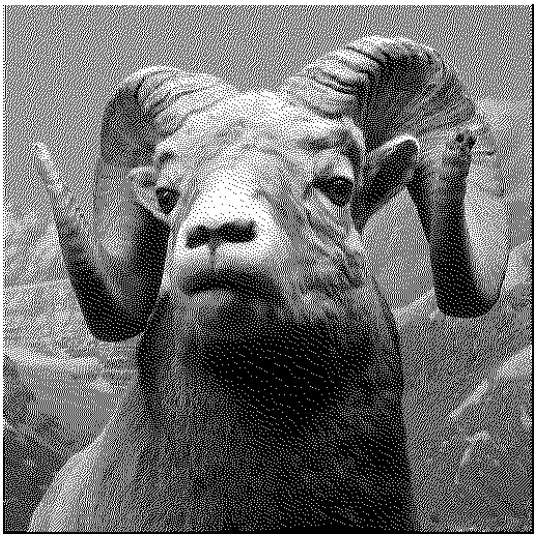
Procedure 1 Conventional Floyd-Steinberg error diffusion

1: **Input:** I_{origin} [a given 8-bit greyscale image]
2: **Output:** I [an 8-bit black and white image for halftoning]
3: $I \leftarrow I_{origin}$
4: **for** each y from top to bottom **do**
5: **for** each x from left to right **do**
6: $oldpixel \leftarrow I(x, y)$
7: $newpixel \leftarrow$ find_closest_color($oldpixel$) [White or Black (255 or 0)]
8: $I(x, y) \leftarrow newpixel$
9: $e_{xy} \leftarrow oldpixel - newpixel$
10: $I(x+1, y) \leftarrow I(x+1, y) + 7/16 \times e_{xy}$
11: $I(x-1, y+1) \leftarrow I(x-1, y+1) + 3/16 \times e_{xy}$
12: $I(x, y+1) \leftarrow I(x, y+1) + 5/16 \times e_{xy}$
13: $I(x+1, y+1) \leftarrow I(x+1, y+1) + 1/16 \times e_{xy}$
14: **end for**
15: **end for**

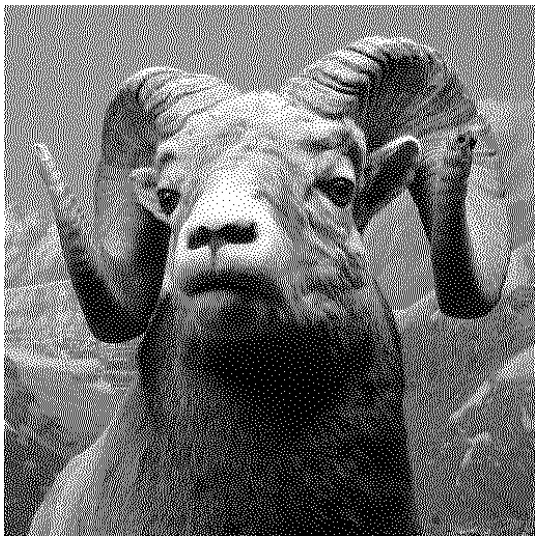
its new pixel is error e_{xy}. For each pixel, the error e_{xy} is distributed to the neighboring pixels in a mask according to distribution coefficients $\frac{1}{16}\begin{bmatrix} & * & 7 \\ 1 & 3 & 5 \end{bmatrix}$. The lines from Line 10 to Line 13 in Procedure 1 show that, for each pixel, the distribution of error affects four nearby pixels. Since Floyd-Steinberg Error Diffusion (FS ED) runs in a raster scanning order, the pixels above and to the left have been processed already. The diffusion does not affect the pixels that already have been ouput. The goal of FS distribution is to maintain the quantization error close to zero over the image.

Figure 3.2 (b) is processed by Floyd-Steinberg error diffusion. It includes several artifacts such as the diagonal lines below the sheep head (see the very dark area of the neck). Ostromoukhov proposed a variable-coefficients error-diffusion algorithm, which employed a serpentine processing path, distributing errors to three instead of four neighbors. Ostromoukhov carefully studied the 256 input intensity levels and introduced 256 coefficient sets for the 256 levels. His method could approximate "blue noise" properties of the distribution and the visual results look more appealing than

(a) Original image



(b) Floyd-Steinberg error diffusion



(c) Ostromoukhov's error diffusion

(d) Contrast-aware halftoning

Figure 3.2: Comparisons with traditional error diffusion. Image resolution is 512 by 512.

those from the Floyd-Steinberg error diffusion algorithm. A halftoning example produced by Ostromoukhov's method is shown in Figure 3.2 (c). The diagonal patterns around the sheep neck are gone. However, both the Floyd-Steinberg error diffusion algorithm and Ostromoukhov's method employed fixed coefficients and their results missed many structural details such as the edges for eyes and the hair, and the texture

of the horns.

Our proposed method is based on the classic error diffusion algorithm and takes full advantage of the tone matching property from error diffusion; our innovation is to distribute error through contrast-aware weights within a normalized mask. In addition, instead of using raster scanning order, the variant of our basic contrast-aware method adopts a flexible dynamic priority scheme to improve fine texture details further. Figure 3.2 (d) demonstrates that our result shows the eyes and the texture of the horns clearly, and vividly expresses the hair. Most error diffusion methods do not have good structural preservation. Our contrast-aware framework outperforms previous halftoning methods in many aspects. The comparisons will focus on structure-aware halftoning methods [15, 97, 139, 192].

3.3 Basic Contrast-Aware Error Diffusion

The core idea behind our basic Contrast-aware Error Diffusion (CED) is to preserve structural details. The goal is to have a contrast-aware distribution of primitives, in which we can preserve image contents. In order to clearly display content in a distribution, we can raise the difference in density between two sides of an edge by having more black pixels in the dark region and fewer black pixels in the bright region. We can further increase the difference in number by adjusting the parameters to have strong contrast.

The basic CED uses raster scanning order and proceeds pixel by pixel. When a pixel is processed, our system must determine which color (black or white) should be chosen. If the input intensity value of the pixel is closer to black, the outcome for that pixel is black; otherwise, white. After quantization, the algorithm calculates the error between the original intensity and the chosen intensity. The error is distributed to nearby pixels beneath a mask. So far the basic algorithm executes in the same way as the previous FS error diffusion method.

We want darker pixels to be more likely to be set to black, while lighter pixels are more likely to be set to white. After the error is diffused to the neighboring pixels, those pixels receive either positive error or negative error, which changes the original intensities of those pixels to new values. Negative error reduces a pixel's intensity value; positive error increases a pixel's intensity. After being added negative

or positive error, a pixel can have a large amount of change in intensities. If this change makes a pixel with intensity below 128 become above 128 and a pixel with intensity above 128 become below 128, the decision on the output color for this pixel is opposite to the decision before error diffusion. Figure 3.3 shows how the basic contrast-aware error diffusion changes a uniform region. The intensity value of a neighboring pixel increased from adding positive error means that the pixel is lightened; the reduced intensity from adding negative error means the pixel is darkened. In simple terms, a black output indicates that the surrounding area should be lightened to compensate to the positive error; a white outcome indicates that the area is too light already and the neighborhood should be darkened. The contrast policy in diffusion is that when positive error is diffused to nearby pixels in a mask, the darker pixels absorb less positive error to avoid being lightened and the lighter pixels absorb more positive error to enhance being lightened. This diffusion strategy biases the result such that pixels already lighter than their neighbors, become even lighter, while darker pixels remain dark. Conversely, negative error, which causes darkening, is distributed preferentially to dark pixels, making them even darker. The outcome of error diffusion for a region with different intensities is illustrated in Figure 3.4.

Uniform Region

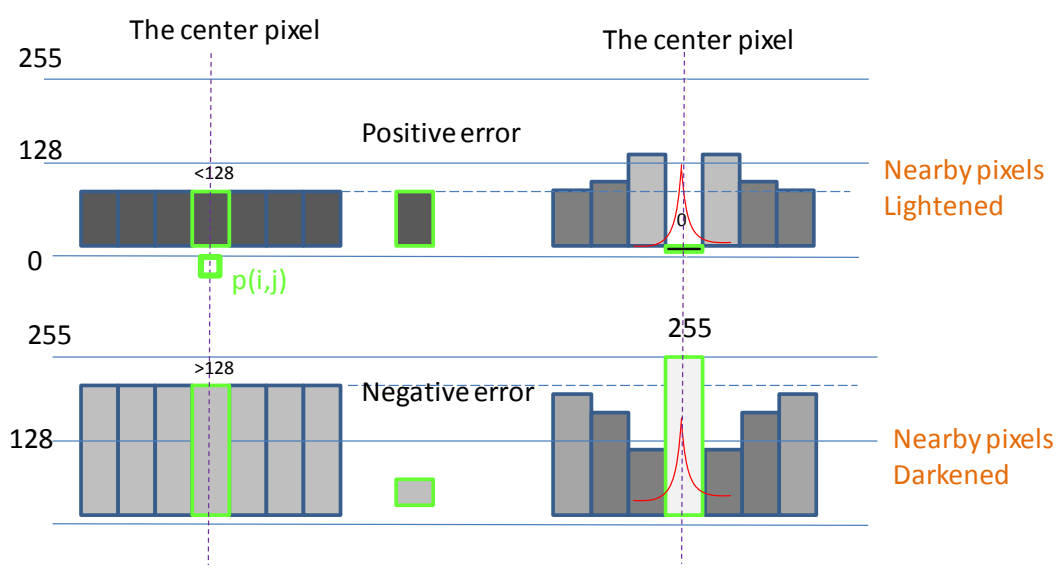


Figure 3.3: Uniform region: the effect on the neighboring pixels after error diffusion.

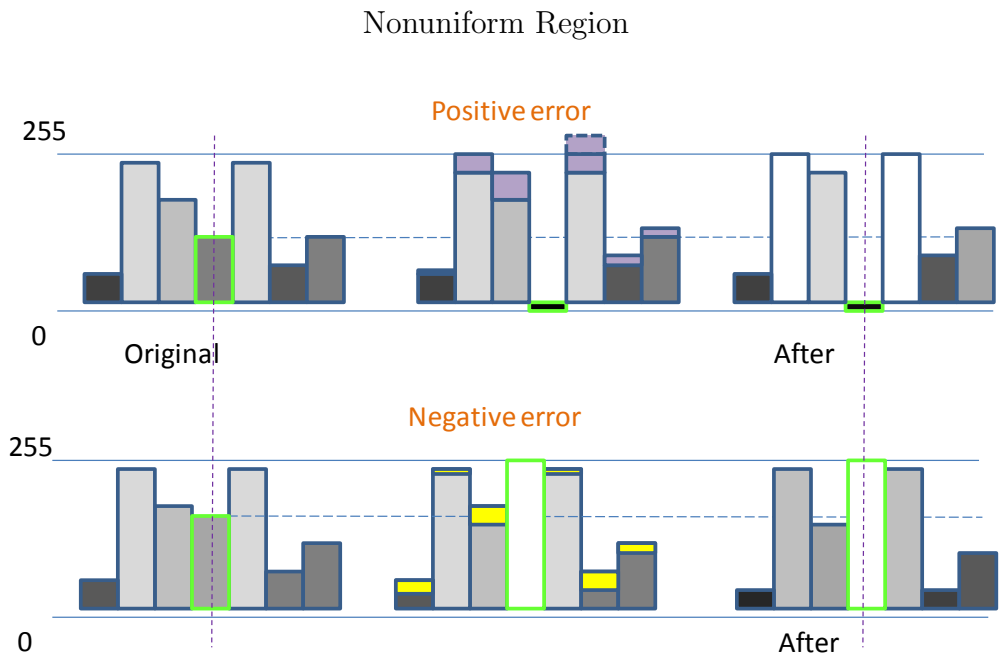


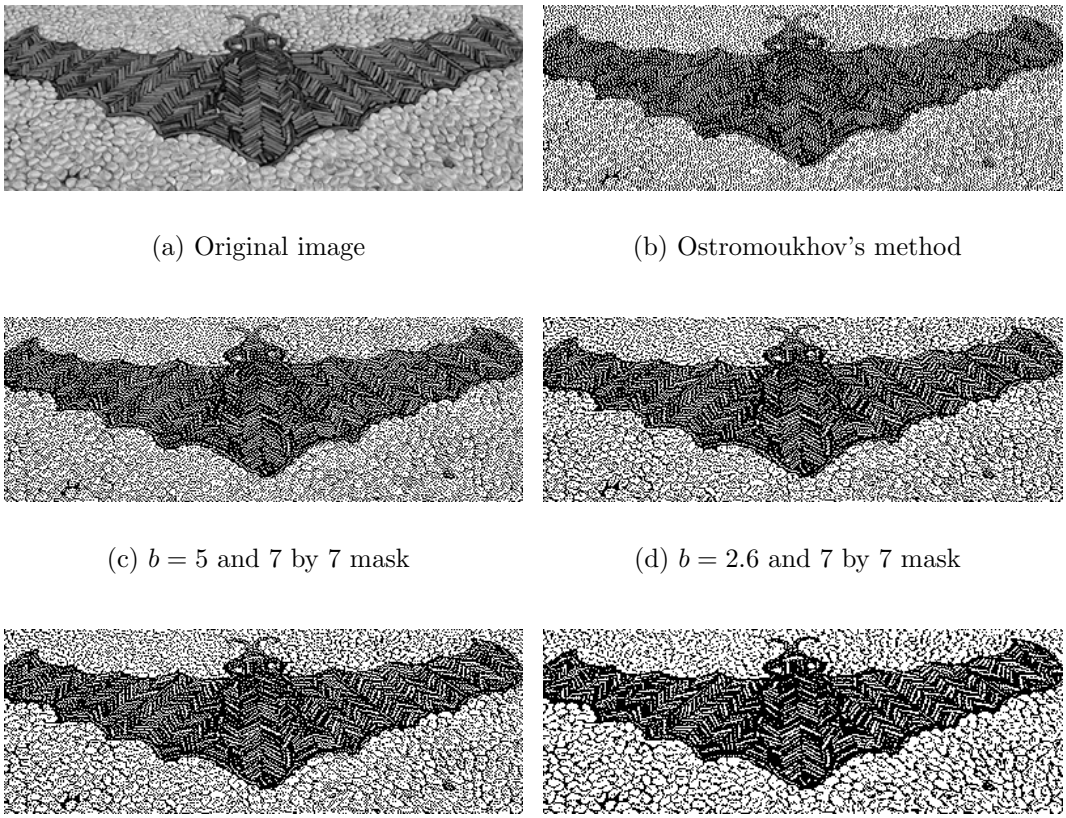
Figure 3.4: Nonuniform region: the effect on the neighboring pixels after error diffusion.

Our process of diffusing error to nearby pixels employs a circular mask centred on each pixel. The distribution coefficients of the error are normalized weights, $\hat{w}_{mn}$, which are calculated such that the sum of all $\hat{w}_{mn}$ is unity. After normalization, each pixel receives error in proportion to its weight. We compute the non-normalized w_{mn} as follows:

$$\text{if } error > 0, w_{mn} = \frac{I_{mn}}{(r_{mn})^b} \tag{3.1}$$

$$\text{else, } w_{mn} = \frac{(255 - I_{mn})}{(r_{mn})^b} \tag{3.2}$$

where we use I_{mn} to represent the intensity of pixel m, n beneath the mask, r_{mn} for the distance of pixel m, n from the mask centre, and b as a positive parameter. The intensity values of an image have a range of $[0, 255]$. Figure 3.5 shows how the structure preservation is affected by parameter choice of b and the mask size. We can barely see the textures in Figure 3.5 (b) generated by Ostromoukhov's method. From Figure 3.5 (c) to (d), we use the same size of masks and reduce b. The edges in Figure 3.5 (d) are clearer and the tone is less matched than those in Figure 3.5 (c). Increasing the size of masks from Figure 3.5 (d) to (e) enhances the edges as



(a) Original image

(b) Ostromoukhov's method

(c) $b = 5$ and 7 by 7 mask

(d) $b = 2.6$ and 7 by 7 mask

(e) $b = 2.6$ and 15 by 15 mask

(f) $b = 1.6$ and 15 by 15 mask

Figure 3.5: (c) - (f): Different parameter values affect the structure and the tone preservation.

well. The explanation of good preservation of structure by using a large mask is that the process involves more pixels and the contrast-aware mask affects more pixels at each time. Within a particular diffusion step, CAH amplifies existing intensity differences between the affected pixels. Local structures in the image, of size no larger than the mask, can thus be preserved through each step of error diffusion. Larger masks are able to preserve larger-scale structures. A small b value, for example, shown in Figure 3.5 (f), increases the influence of the distance to the center and further maintains the structure preservation. The parallel texture and the pebbles in Figure 3.5 (f) are much clearer than those in Figure 3.5 (b), (d), and (e). However, if b is very small, the resulting images hardly represent the tone matching. A small b brings a large increase in difference of intensities when doing error diffusion. Areas with intermediate tone after receiving error using a small b become either closer to

black or closer to white than using a large b. In the end, the overall loss of tone for the entire image becomes huge, which does not imitate the original tone any more. Figure 3.5 (f) looks lighter than (b), (d), and (e). We suggest $b = 2.6$ with a 7 by 7 mask. This suggestion for the parameter b and the mask size is based on many experiments we have done on a variety of images including photographs and illustrations with a variety of image contents such as natural scenes, portraits, buildings, animals, trees, and cars. All our halftoning results using the suggested parameter values on the tested images are good in structural preservation. Our results are obtained under a couple of seconds. This thesis only shows a small number of our halftoning results.

Error diffusion is applied to only unprocessed pixels within the mask region. Weights for processed pixels are zero, which means error diffusion will not affect processed pixels. We compute the normalized weights using dividing by the sum of weights for all neighboring pixels that were not previously set to their final values; such pixels are indicated as *done*.

$$W_{total} = \sum_{(m,n) \in neighbors \setminus done} w_{mn} \tag{3.3}$$

$$\hat{w}_{mn} = \frac{w_{mn}}{W_{total}} \tag{3.4}$$

In the equation, W_{total} is the sum of weights. No distribution happens when W_{total} is zero, which means an isolated pixel. Otherwise, a portion of the error is received in intensity adjustments:

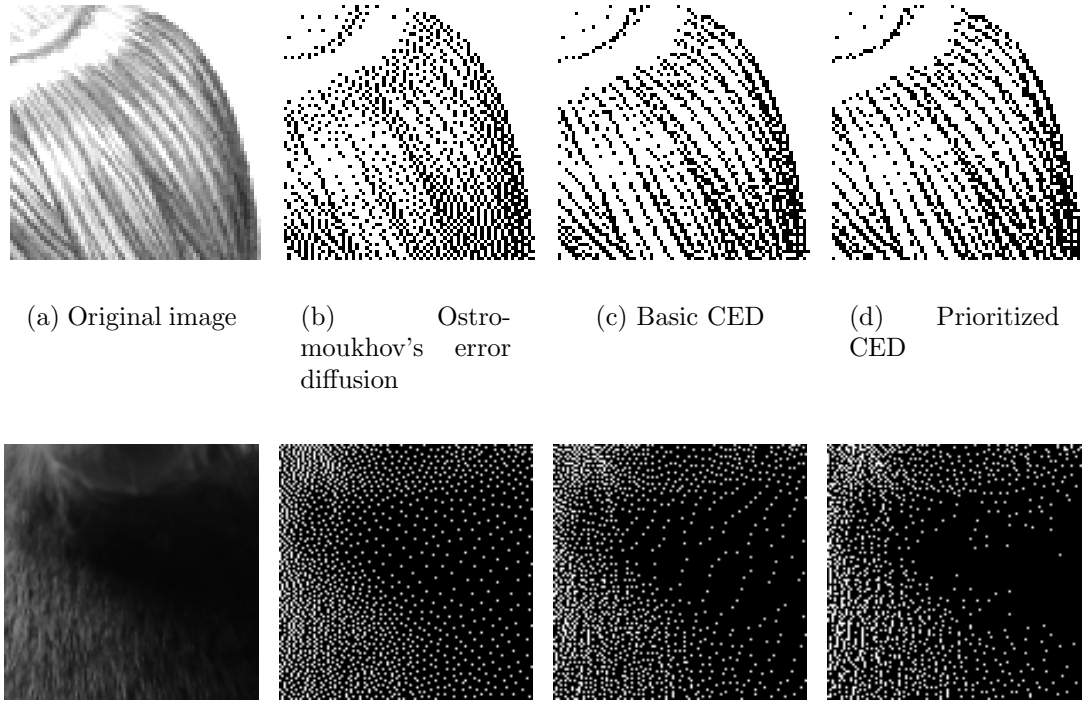
$$I_{mn} = I_{mn} + error \times \hat{w}_{mn}. \tag{3.5}$$

Notice that the intensity adjustment in Equation 3.5 may result in overflow or underflow, above 255 or below zero. In such cases, we clamp the pixel to 0 or 255 and the excess due to clamping is added to the residual. In the case of isolation, the error is not distributed to the neighborhood and the residual error is accumulated forward. If all residual error is positive, the residual error would become a large positive value; if all residual error is negative, the residual error would become a large negative error. In both cases, the overall loss in tone is a large amount and the tone difference between the halftoned image and the original image becomes large. In the process, the algorithm carries residual error forward to the next pixel, helping reducing the residual error.

Figure 3.6 (a) is an illustration with lots of texture including weak edges. Error diffusion by Ostromoukhov's method produces the halftoning result in (b) with rare edges preserved. In Figure 3.6 (c), basic CED preserves many weak edges. Our basic CED scheme performs better than Ostromoukhov's method and the FS ED method in structure preservation. However, the raster scanning order is too inflexible. Raster scanning order processes pixels from left to right and from top to bottom. For a pixel not in the image border, its above and left pixels always have been processed; and this pixel will be forced to receive error from the above and the left. After a few times of diffusion, the accumulated error could be high. The intensity of the pixel might be changed a lot after receiving the error, which means a dark pixel would become a bright pixel or a bright pixel would become a dark pixels. For example, when all error accumulated from previous pixels are positive, the intensity change for this pixel is high and it is easy for this pixel to become a white pixel. For pixels at weak edges, the difference among intensities of pixels are small. A large change in intensity after error diffusion in raster scanning order easily causes pixels to move toward to one of the two extreme intensities, black or white. When thresholding pixels with a small difference after diffusion, our process outputs either black or white, ignoring the small difference. Inflexible ordering causes weak edges to be missed. The order introduces visible artifacts. The raster scanning order in the basic CED creates the same visual artifacts as previous conventional error diffusion methods. The bottom row of Figure 3.6 (g) shows that some pixels are organized into spurious. The line patterns are not structures from the original image. We propose Prioritized CED to resolve the artifacts.

3.4 Variant with Dynamic Priority

We introduce priority-based contrast-aware error diffusion, Prioritized CED, to resolve the disadvantages of the basic CED. Priority-based CED breaks the inflexible raster scanning order and uses priority-based order. Pixels close to black are more likely to be black and those close to white are more likely to be white. The idea is to have the pixels closer to black or to white processed first, and have the intermediate pixels treated last, after the pixels with non-extreme values have received the error from more extreme-valued neighbors. The central idea of error diffusion is to make



(a) Original image (b) Ostromoukhov's error diffusion (c) Basic CED (d) Prioritized CED

(e) Original image (f) Ostromoukhov's error diffusion (g) Basic CED (h) Prioritized CED

Figure 3.6: The basic CED expresses the weak edges nicely, but there are artificial patterns, which are removed by the Prioritized CED.

up the loss of tone by lowering or raising the intensities of the nearby pixels. The priority scheme fits to minimize the loss of tone. When the intensity of a pixel is 0, the process of error diffusion for this pixel outputs a black pixel; when the intensity of a pixel is 255, the process of error diffusion outputs a white pixel. In both cases, the error is zero. This priority strategy processes the pixels closer to two extremes earlier. The absolute error in each step is the minimum error from all pixels, providing the minimum loss of tone after outputting a black pixel. Pixels closer to white and black are assigned higher priorities, which fits exactly our purpose of allowing pixels close to black or white to be processed first.

We maintain the priorities of pixels and process them in priority order, where pixels with higher priority are those closer in intensity to extreme values (white or black). Notice that as error is diffused from processed pixels, a pixel's intensity and hence priority can change. The calculation of the priority $p(x, y)$ at position (x, y) is

shown in Equation 3.6.

$$p(x,y) = \begin{cases} 255 - I(x,y) & \text{if } I(x,y) < 128 \\ I(x,y) & \text{otherwise.} \end{cases} \tag{3.6}$$

In order to implement this idea, a small modification to the basic procedure is enough. The algorithm keeps a priority heap, with all pixels sorted by their intensity distance to black or to white. Initially, the heap stores all pixels. Each step pops up the pixel with highest priority: of all remaining pixels, the one with the smallest distance to black or white. CED is applied to this one pixel, assigning the final color and distributing the error to its neighborhood. After the error distribution, many nearby pixels will have their intensities changed, and correspondingly their priorities revised as well. When the process finds a pixel with a new value, it pushes the up-to-date value onto the heap; in the implementation, there will be stale values for this same pixel in the heap, which the algorithm can discover and discard by checking the popped up pixel to verify that its priority is up to date. The algorithm continues until the heap is empty, at which point all pixels will have been assigned a final value.

In the dynamic priority variant, the best tradeoff between quality and speed is using $b = 2$ and a mask size of 7 by 7. Lowering b enhances the structure preservation by presenting high contrast image. The priority scheme breaks the spurious patterns to seek a better output of black pixels. The broken patterns hide some edges and are not helpful for displaying the major structure. However, Figures 3.7 and 3.8 show that better structure detail is provided by priority-based scheme than by the basic CED. You can take a look at the "snail" example in Figure 3.8, where the result from basic CED misses some weak edges (see the inner circle on the left side). The result for the "snail" example by Prioritized CED improves the missing structure. We tested both basic and Prioritized CED on a wide range of image examples. Notice that the structure preservation from the Prioritized CED is better than that from the basic CED. The raster scanning order used by the basic CED has limitation on displaying weak edges, while the priority scheme improves weak edges. The reason is that, after a diffusion is finished, the next pixel to be processed could be anywhere in Prioritized CED and the previous error diffusion has little chance to influence the future diffusion. The little influence avoids changing a pixel's intensity too much on

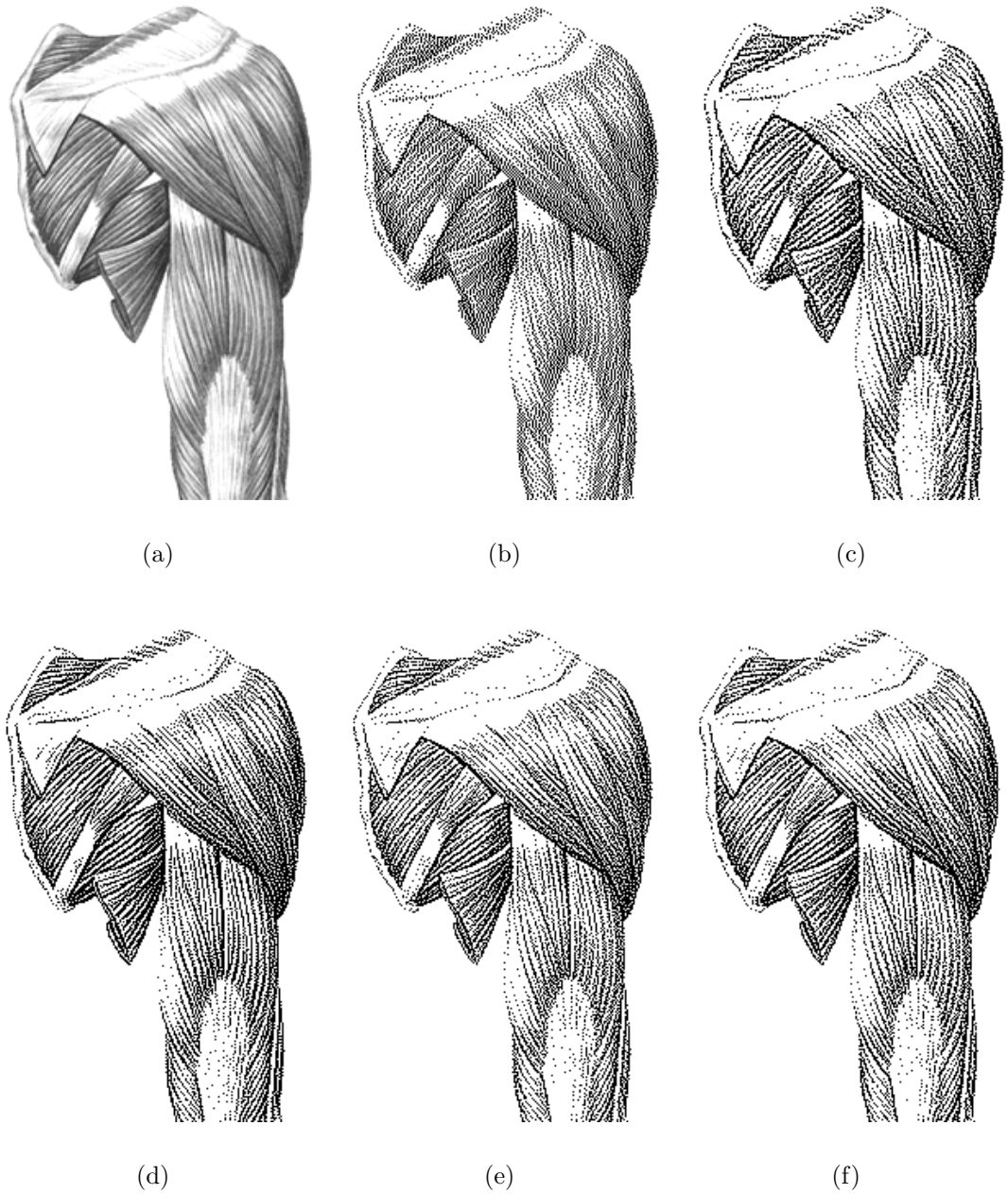


Figure 3.7: Illustration "arm". (a) Original image; (b) Ostromoukhov's error diffusion; (c) SAH [139] (around 26 seconds); (d) SAED [15]; (e) basic CED (0.19 seconds); (f) Prioritized CED (0.48 seconds).

weak edges and in the end keeps lots of weak edges. In halftoning a uniform region, the pixels within the mask will have their priorities reduced after distributing error; hence, those modified pixels are processed later. The next pixel to be chosen will be a pixel outside of the current mask, not a neighboring pixel. Choosing a distant pixel avoids clustering, somewhat simulating Poisson disc behavior and producing a

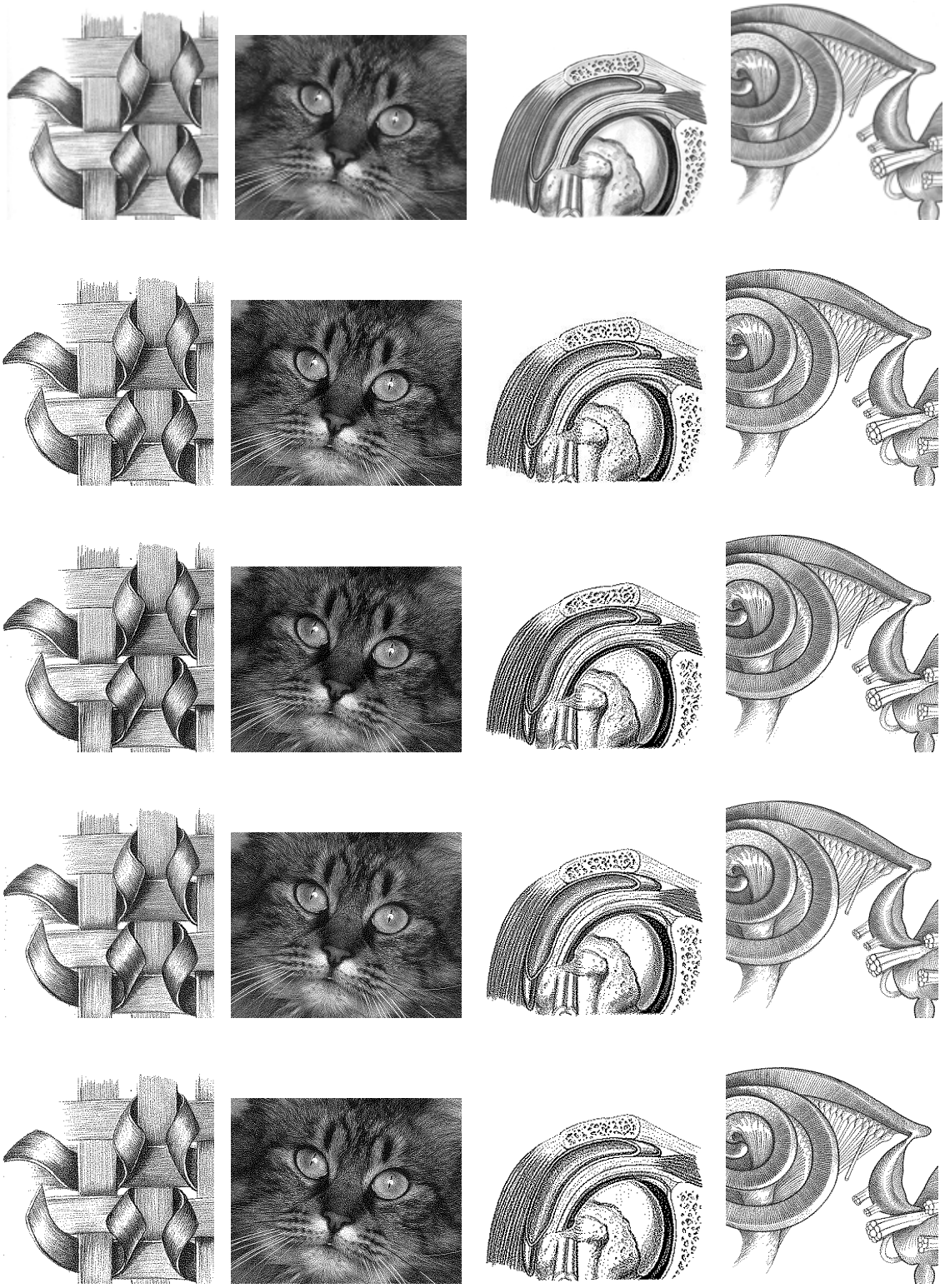


Figure 3.8: “ribbon”, “cat”, “knee”, and “snail” with image resolution 445 by 377, 367 by 373, 269 by 203, and 384 by 399 respectively. From top to bottom: original image; structure-aware halftoning [139]; structure-aware error diffusion [15]; basic CED; Prioritized CED.

better spatial distribution. The Poisson disc behavior can approximate "blue noise" properties [174], whose visual appearance is characterized by evenly placed primitives. The error in a mask is distributed in such a way as to preserve contrast. An up-to-date locally prioritized order allows us to use all the available information in deciding which pixel to be treated next, including the history of distributed errors. Processing in this way empirically results in superior detail preservation.

In Figure 3.9, the visualization of the prioritized orders shows that the edges of

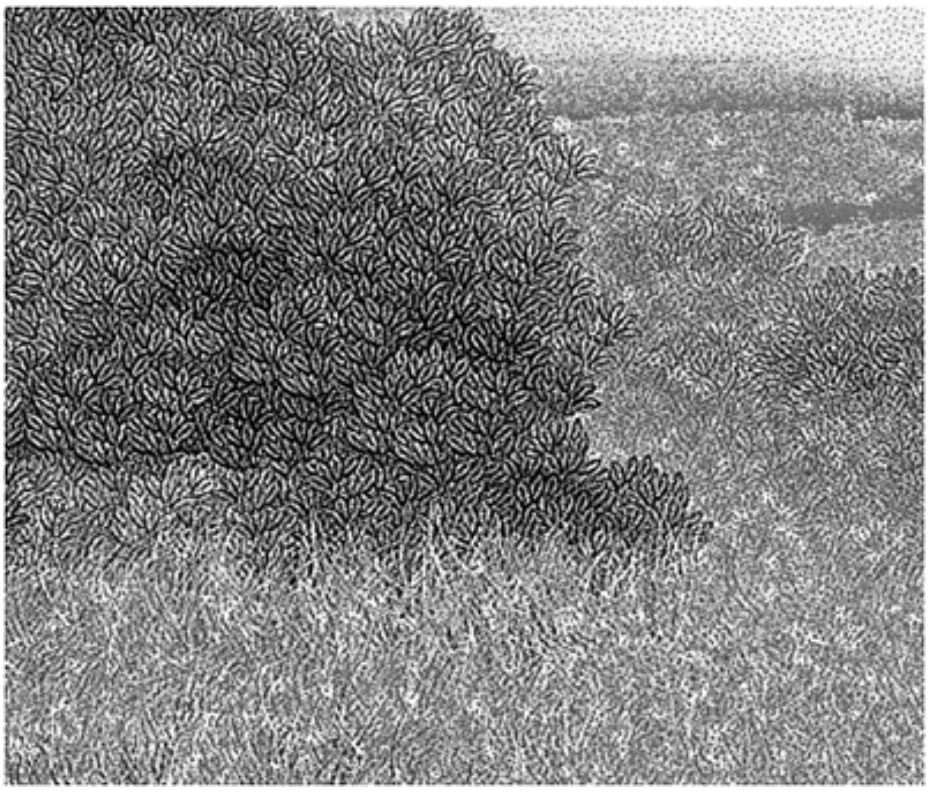


Figure 3.9: Visualizing the order after using Prioritized CED. The first pixel is marked as black and the last pixel is marked as white. The intensities of in between pixels are proportional to their orders in the range of [0, 255].

the leaves are darker than the background sky, indicating that pixels on the edges are processed earlier. We notice a small issue. For example, if we process a uniform region with only one intensity, all priorities will be assigned the same values. If we insert pixels into heap following raster scanning order, the process will pop up a pixel that is one of the neighbors of the previous pixel. In this case for a uniform region, although the priority-based scheme is used, the process runs exactly as the process

in the raster scanning order. To avoid the tie from raster scanning order, before using the priority-based scheme, we add a new step to randomly load all pixels for an original image to break the tie and improve the visual appearance.

3.5 Evaluation

Conventional error diffusion algorithms mainly investigate tone matching and lack structural matching. We compared our halftoning results with the results of previous tone-based error diffusion methods. However, our interest is in comparisons between our method and existing structure-aware halftoning methods. The rest of the chapter will give a detailed evaluation of our halftoning methods to show that the contrast-aware method has advantages over existing structure-aware halftoning methods in many aspects.

Chang et al. [15] stated that their results by structure-aware error diffusion (SAED) have higher tone matching and lower structure matching than the results from structure-aware halftoning (SAH) by Pang et al. [139]. Our major interest was in structure similarity: therefore, most of our results are compared with results from SAH, of higher quality in depicting content than SAED. We provide measurement data in Table 3.1, 3.2, and 3.3, including data for Ostromoukhov's method [134] and Floyd-Steinberg error diffusion (FSED) [39] in order to benchmark against methods not specifically designed for structure. Visual comparisons of our methods with SAH and SAED are shown in Figures 3.8, 3.11, and 3.13.

Measurements of tone and structural similarity use the same calculation as SAH [139] and SAED [15]. We propose a variant from the calculation of tone similarity [166] for the measurement of contrast similarity. Programs for objective evaluations are coded in Matlab. To compare with SAH and SAED, we take the respective authors' evaluation design, smoothing both the original image and the result with an 11 by 11 Gaussian filter, and using the standard deviation $\sigma = 2.0$ for tone measurement, $\sigma = 1.5$ for structure measurement. We use the standard deviation $\sigma = 0.5$ in measuring contrast matching. To have a fair comparison with SAH, the 12 original images we used in the objective evaluations are the images used by both previous papers of SAH and SAED. Those original images have highly-textured contents and weak edges, difficult for many previous halftoning methods to produce. The following

Image	Prioritized CED	Basic CED	SAH	Ostromoukhov's method	FS ED
lion	31.96	34.02	41.12	**43.87**	41.57
mole	34.29	36.81	40.94	**44.83**	44.65
pelican	42.19	43.16	42.59	48.00	**48.05**
portrait	39.29	40.91	38.92	43.61	**44.33**
ribbon	38.70	39.28	35.51	45.96	**46.49**
road	35.33	37.14	38.48	**44.01**	42.82
arm	42.18	41.22	41.20	35.95	**47.67**
bat	36.85	39.28	39.30	39.78	**44.68**
cat	32.86	36.07	33.63	34.71	**44.24**
knee	39.17	38.47	37.64	34.92	**44.87**
snail	40.58	41.25	40.09	37.07	**45.67**
tree	39.14	41.77	39.41	40.85	**48.44**
Average	37.71	39.12	39.07	41.13	**45.23**

Table 3.1: Tone similarity measurement based on PSNR. Bolded numbers are the highest scores of all five methods. The underlined numbers are the better values of our results as compared only with SAH method.

sections show the measurement details.

3.5.1 Tone Similarity

Tone similarity is usually measured by computing the Peak Signal-to-Noise Ratio (PSNR) between an original image and its rendered image in terms of the logarithmic decibel scale [166]. Table 3.1 provides the statistics in PSNR values for tone measurement. PSNR is defined as follows:

$$\text{PSNR} = 20 \times log_{10}(\frac{255}{\sqrt{\text{MSE}}}), \tag{3.7}$$

where MSE is the Mean Square Error (MSE) between the original image and the halftoning result. Basically, the MSE and the PSNR are the same measurement, but the MSE depends strongly on the image intensity scaling. Visually, the same value of MSE in an 8-bit image and in a 10-bit image may have quite different appearances. The PSNR scales the MSE according to the image range in order to avoid this big difference. A small MSE roughly indicates that there is very little difference between the original image and the rendered image, which provides a high PSNR value. The less the difference in MSE, the higher the PSNR, and the closer the halftoned image

is to the original image. Generally, a higher PSNR value should correlate to a higher quality image, but the data have shown that this is not always the case. For example, the same values of PSNR may not say that the tested images have the same quality, because they may come from adding noise to images or blurring images. The PSNR measurement does not consider any important feature, but only calculates the tone difference. However, using the PSNR is popular for measuring tone matching because it is easy to understand and fast to calculate while giving reasonable results. The researchers who proposed the SAH and SAED approaches used this metric to measure tone matching.

From the comparisons given in Table 3.1, we see that eight (underlined) out of twelve results by the basic CED attain higher tone similarity than SAH. The average score of basic CED is higher than that of the SAH method. Five (underlined) out of twelve results from the Prioritized CED approach attain higher scores than SAH. The average score of Prioritized CED is lower than that of SAH. One explanation of a little lower tone matching is that more black pixels contribute to draw subtle details than strict tone matching would allow. Figure 3.10 shows the comparison for halftoning the ramp intensity with recent halftoning methods. We notice that the undesirable patterns near intensity 128 in the ramp (the middle part of the ramp image), visible in all previous methods, are eliminated by using the dynamic priority method.

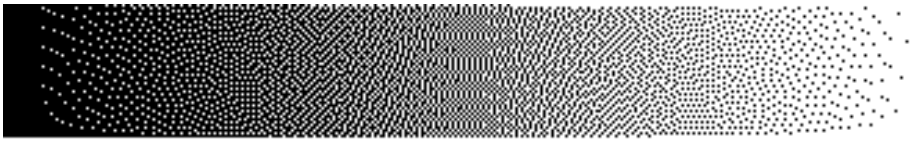
3.5.2 Structure Similarity

Structural information in an image is the key information of objects, including silhouettes, discontinuous, creases, and textures, those with rapid changes in tone values. The values in Table 3.2 show structure similarity measured by using the difference of the structural information between the rendered image and the original image. We employ the Mean Structural SImilarity Measure (MSSIM) [181] to evaluate structural quality as follows,

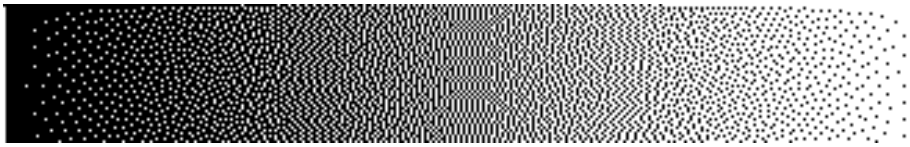
$$\mathrm{MSSIM}(I, I_h) = \frac{1}{N}\sum_{i=0}^{N} \mathrm{SSIM}(x, y), \tag{3.8}$$

where SSIM is the overall structure similarity and I and I_h are an original image and its rendered image. The value of MSSIM ranges from 0 to 1. The higher the MSSIM

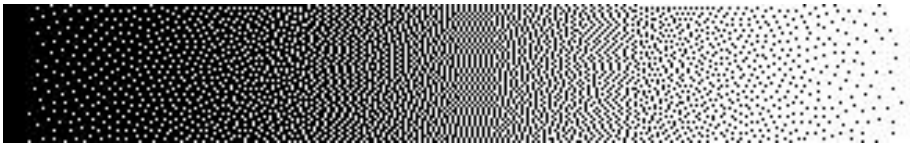
(a) Original image



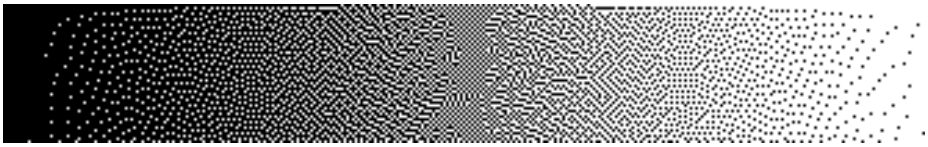
(b) Floyd-Steinberg error diffusion



(c) Ostromoukhov's method



(d) SAH [139]



(e) Basic CED

(f) Prioritized CED

Figure 3.10: Tone quality: comparison for ramp intensity.

Image	Prioritized CED	Basic CED	SAH	Ostromoukhov's method	FS ED
lion	**0.1074**	0.0898	0.0822	0.0553	0.0613
mole	**0.1299**	0.1063	0.1011	0.0616	0.0680
pelican	**0.7853**	0.7691	0.7716	0.7283	0.7435
portrait	**0.3246**	0.2877	0.2745	0.1861	0.2043
ribbon	**0.3576**	0.3292	0.2851	0.2836	0.2921
road	**0.3341**	0.3003	0.2906	0.1879	0.2070
arm	**0.5738**	0.5511	0.5479	0.3845	0.5002
bat	**0.2985**	0.2556	0.2678	0.1617	0.1620
cat	**0.1514**	0.1142	0.1230	0.0673	0.0648
knee	**0.4866**	0.4563	0.4534	0.2951	0.4359
snail	**0.4682**	0.4386	0.4475	0.3803	0.4090
tree	**0.1916**	0.1551	0.1539	0.0317	0.095
Average	**0.3508**	0.3211	0.3166	0.2353	0.2703

Table 3.2: Structure similarity measurement based on MSSIM codes proposed by Wang [181]. Bolded numbers are the highest scores of all five methods. The underlined numbers are the better values of our results as compared only with SAH method.

values, the higher the structural similarity. SSIM measures the difference in local structure similarity, calculated as follows,

$$\mathrm{SSIM}(x, y) = \frac{(2\mu_x\mu_y + k_1)(2\sigma_x\sigma_y + k_2)}{((\mu_x)^2 + (\mu_y)^2 + k_1)((\sigma_x)^2 + (\sigma_y)^2 + k_2)} \tag{3.9}$$

where μ is the Gaussian weighted mean intensity; σ is the standard deviation; and k_1 and k_2 are small positive constants avoiding singularity. Suppose x and y are two nonnegative image signals; then μ_x and μ_y indicate the mean intensity of the signals and σ_x and σ_y indicate the standard deviation as an estimate of the signal contrast showing high structure features. The measurement combines the similarity with three components: luminance, contrast, and structure. Wang et al. [181] provided a detailed calculation. The MSSIM in practice is very robust and is used in measuring structural similarity between a rendered image and an image reference in image processing. Previously, structure-aware methods by Pang et al. [139] and Chang et al. [15] also used this metric to evaluate their results in terms of structural quality. The MSSIM values in Table 3.2 show that all twelve results (underlined) from the Prioritized CED method achieve higher scores in this test than those from SAH and eight (underlined) out of twelve results from our basic CED are better than those from SAH. Both

average scores of basic and Prioritized CED are higher than those of SAH.

Our method has the ability to display weak edges. In Figure 3.11, our results show the leaves very clearly; the deepness of the shadows is strong; and the faint halo along the right-hand edge of the main bush is expressed well. It is even possible to detect subtle texture in the far away bush, which is not shown in the result from SAH. Further improvement of edge preservation is due to prioritized order. SAH by Pang et al. [139] is an optimal and iterative approach. Presumably, given more time, SAH would be able to produce better structure similarity than our method. However, using the optimization scheme Pang et al. described, it may take a prohibitively long time to find out significant gains in structure quality. The reason is that in each iteration the attempt is randomly taken in a pool, in which contains many unprocessed pixels. The outcome of the attempt is unpredictable in terms of tone and structural quality. It is possible for SAH to take a very long time to have a good attempt. In comparison, our basic CED visits each pixel once. After all pixels in an image are visited, the process is finished. Prioritized CED takes a longer time than basic CED but still runs in a single pass. Extra time is spent on maintaining the data structure. The update of the priority orders makes the processing time of Prioritized CED longer.

3.5.3 Contrast Similarity

The evaluation data shows that our halftoning results have reasonable tone matching and very high structural matching. We further introduce contrast similarity to validate our proposed halftoning methods. Image processing has approaches to measure luminance contrast [147], which is formulated as $\frac{LuminanceDifference}{AverageLuminance}$, to describe perceived luminance difference in a small region. The calculation follows the Weber-Fechner law [147], saying that the magnitude of a subjective sensation increases proportionally to the logarithm of the stimulus intensity. The formula indicates that the average luminance is important in calculating contrast. If the average luminance is low, even a small alteration matters. Practically, the arguments by Neumann et al. [129] and Matkovic et al. [119] said that the measure for local contrast limits the evaluation for the quality of rendering [151] that only locally operates a contrast sensitivity function [114] in frequency space. The calculation of global contrast is therefore proposed.

(a) Original image

(b) Ostromoukhov's method

(c) Structure-aware halftoning [139]

(d) Structure-aware error diffusion [15]

(e) Basic CED

(f) Prioritized CED

Figure 3.11: Illustration "tree". Image resolution is 445 by 377.

We propose to measure contrast quality based on local contrast. Contrast similarity is measured as follows: we calculate the local contrast pixel by pixel for both the original image and the result; the calculated values are stored in two maps as two contrast images; we then find the mean squared error (MSE) between the two contrast images; and the PSNR value is used also for the contrast similarity. The Contrast PSNR (CPSNR) is calculated in same the way as we did for the tone similarity. Computing the local contrast is proposed by Matkovic et al. [119], taking into account human perception. The perceptual luminance $L_{x,y}$ at position (x, y) is computed by applying a gamma correction to modify the linear luminance, where γ is 2.2, as suggested by Matkovic et al. [119]. The local contrast $lc_{x,y}$ is an averaged value of the local difference among the neighbors. The equations are as follows:

$$L_{x,y} = 100 \times \sqrt{(g_{x,y})^{\gamma}} \tag{3.10}$$

$$lc_{x,y} = \frac{\sum_{(m,n)\in\text{Neighbors}} |L_{m,n} - L_{x,y}|}{4} \tag{3.11}$$

where $g_{x,y}$ is a normalized grayscale level $\in [0, 1]$, (x, y) represents the position of the center pixel, and we use four-connected neighbors. Similar to the PSNR for tone measurement, the CPSNR has a higher value if the halftoned image is closer to the original image in terms of contrast. The CPSNR measurement has the same limitation as the PSNR measurement, because the same CPSNR values may have very different visual appearance. As can be seen in Table 3.3, both our algorithms outperform other methods on this metric.

3.5.4 Blue Noise Property

Blue noise is a particular distribution in frequency domain with no spikes in energy and minimal low frequency components. Dots possessing blue noise properties are distributed evenly. People commonly agree that halftoning results with the blue noise property are visually appealing [174]. Quality is measured by the Radially Averaged Power Spectrum Density (RAPSD), a widely used measure for estimating whether an image has the blue noise property. We first calculate the estimated 2D power spectrum $P(f)$ by finding the Fourier amplitude spectrum and using Bartlett's method [5] of

Image	Prioritized CED	Basic CED	SAH	Ostromoukhov's method	FS ED
lion	13.23	**13.32**	12.56	12.23	12.49
mole	11.72	**11.94**	11.12	10.64	10.95
pelican	**18.03**	17.84	17.54	16.17	16.69
portrait	**12.73**	12.68	12.11	11.24	11.71
ribbon	13.26	**13.31**	12.26	12.04	12.37
road	12.24	**12.25**	11.62	10.55	11.10
arm	14.51	**14.57**	13.97	12.22	13.58
bat	**12.06**	11.99	11.95	11.62	11.04
cat	11.62	**11.70**	11.16	10.40	10.63
knee	**13.88**	13.85	13.55	11.40	12.48
snail	13.75	**13.89**	13.46	12.15	12.67
tree	11.72	**11.90**	11.13	10.02	10.74
Average	13.23	**13.27**	12.70	11.72	12.20

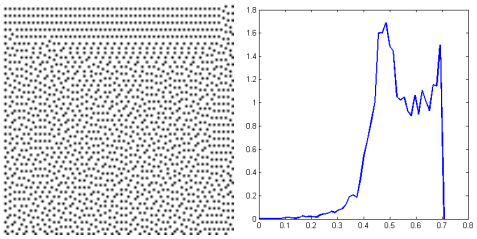
Table 3.3: Contrast similarity measurement based on contrast PSNR. Bolded numbers are the highest scores of all five methods. The underlined numbers are the better values of our results as compared only with SAH.

averaging periodograms. The RAPSD ($P_r(f_r)$) [174] is then defined as follows,

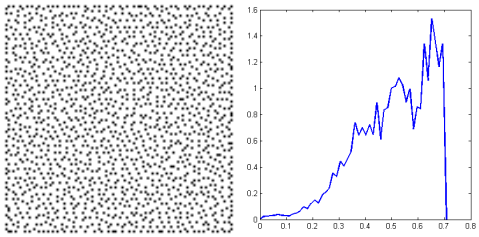
$$P_r(f_r) = \frac{1}{N_r(f_r)} \sum_{i=1}^{N_r(f_r)} P(f) \tag{3.12}$$

where in the Fourier domain f_r represents the radial frequency and $N_r(f_r)$ means the number of frequency samples within the annular ring. The RAPSD is the average of all possible directional power spectra, which is direction-independent. Using the RAPSD, we can conveniently view and compare information contained in 2D spectra in 1D.

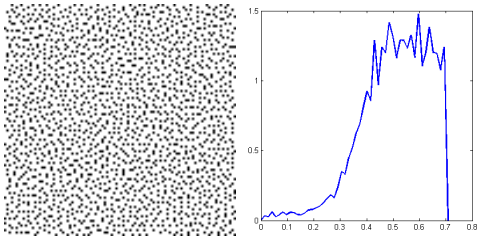
In Figure 3.12, we show a visual analysis of the blue-noise properties of black pixel distribution. Figure 3.12 (a) is a result generated by basic CED without breaking ties and its RAPSD; it includes undesirable patterns. We created Figure 3.12 (b) by using Prioritized CED; it is of better quality but the RAPSD has no steep jump from low to high frequencies, lacking the blue-noise property. The problem arises because the original image has a uniform color, having the same priority values for all pixels. The output order of pixels is tightly connected to the order of inserting pixels. The original order of inserting pixels is raster scanning order, with concomitant artifacts. Figure 3.12 (d) shows the halftoning result by adding a random tie-breaking



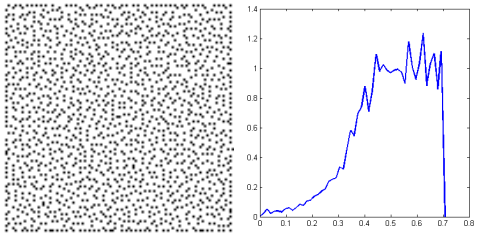
(a) Basic CED



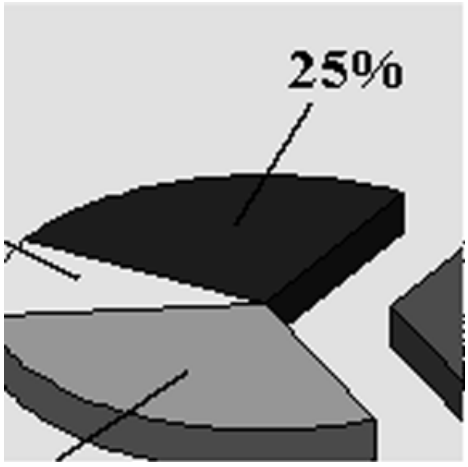
(b) Prioritized CED



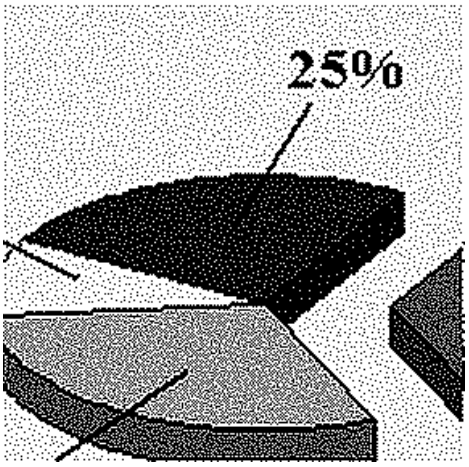
(c) SAH



(d) Prioritized CED with tie-breaking



(e) A cartoon image



(f) Improved Prioritized CED

Figure 3.12: Analysis for blue noise property. We show halftoned images and the RAPSD plots for four halftoning methods: (a) basic CED; (b) Prioritized CED; (c) SAH; (d) Prioritized CED with random tie-breaking of equal priorities. Next, application to real image: (e) input cartoon image; (f) halftoning of (e) with random tie-breaking.

Image	Prioritized CED	Basic CED	SAH	Ostromoukhov's method	FS ED
PSNR	37.71	39.12	39.07	41.13	**45.23**
MSSIM	**0.3508**	0.3211	0.3166	0.2353	0.2703
CPSNR	13.23	**13.27**	12.70	11.72	12.20

Table 3.4: Summary of average scores for measurement of PSNR, MSSIM, and CPSNR. Bolded numbers are the highest scores of all five methods. The underlined numbers are the better values of our results as compared only with SAH method.

mechanism: in its RAPSD plot, the high frequency region goes flatter than that from basic CED, which indicates that the blue-noise property is improved. Figure 3.12(c) shows that the plot of the improved result has similar quality to that of the result from SAH. Figure 3.12 (f) demonstrates how tie-breaking mechanism is applied for a cartoon-style image. The tie-breaking scheme randomizes all pixels before inserting them to the system. In this way, even pixels with the same priority will be processed randomly. After this improvement for Prioritized CED, halftoning results have a distribution of black pixels evenly, which resembles blue noise.

3.5.5 Discussion and Analysis

We have reported different aspects of the quality of our halftoning methods, including tone similarity, structure matching, and contrast similarity. Higher values for each measurement indicate higher matching. PSNR and CPSNR are based on the MSE, which is not robust. It may happen that halftoned images with good measurements of PSNR and CPSNR have bad matching visually. MSSIM is robust. The results from MSSIM can give us clear conclusions regarding structural matching. As a result, PSNR and CPSNR provide a rough comparison and we rely on MSSIM to make our conclusion. Table 3.4 summarizes the average scores for measurements of PSNR, MSSIM, and CPSNR on 12 images we tested. The overall scores for both our basic and Prioritized CED are much higher than existing structure-aware halftoning regarding aspects of structural and contrast preservation. All measurements for both our basic and Prioritized CED provide better scores than those for SAH except that the PSNR value on Prioritized CED is lower. The high matching scores in structure come from the strong contrast in describing edges. An increase of the difference of densities of black pixels helps show object edges clearly. This adjustment of the difference is

Mask size	5×5	7×7	9×9	11×11	13×13
Time (basic CED, seconds)	0.27	0.492	0.775	1.225	1.706
Time (Prioritized CED, seconds)	2.202	2.955	3.12	3.608	4.047

Table 3.5: Time for a 512×512 image with different mask sizes.

connected to contrast and modifies the original tone distribution in order to maintain good structure preservation. The modification on tone lowers the tone matching with the original reference. This happens to SAH also. SAH has a higher score on MSSIM but has a lower score on PSNR, as compared with FS ED algorithm. Based on current suggestions on parameters, if we would like to have very high structural matching, we may lose tone matching. High values for both tone and structural matchings are difficult to obtain at the same time. We can conclude that our halftoning methods have better quality in terms of structural preservation, based on the measurement on MSSIM in Table 3.4.

We only show several images in this chapter, which allows us to have a comparison with other previous halftoning methods. Beyond those images we showed, we have tested over two hundred images including a varied range of image contents. All provided us good quality in structure preservation. Next we discuss the performance and the use of parameters, and extend our binary halftoning to color halftoning.

Timing: Assume an image has N pixels and our circular mask has k pixels, the basic CED proceeds each pixel with the circular mask and the time complexity is $O(kN)$. For the Prioritized CED, building the heap will take $NlogN$ steps. Then our CAH system will go through each pixel with error diffusion in a neighborhood having k pixels, which takes kN steps. Beyond building and diffusion, we should add the updating time for the modified pixels. Since each pixel will take $logN$ to update and there are N pixels, the total maximum updating time is $NlogN$. The algorithm runs around $2NlogN + kN$ steps in total. The time complexity is $O(NlogN)$. Based on this, we can say that the time complexity is dependent on image size, but independent with image content and k. Table 3.5 provides data for timing information.

Figure 3.5 shows the tradeoff between tone and structural quality visually. To attain the best tradeoff between quality and speed, processing a 512×512 image with a 7×7 mask takes 0.492 seconds for CPU to run the basic CED and 2.955 seconds

for the Prioritized CED to process. Our focus is on comparison of the performance with structure-aware halftoning. Pang et al. [139] reported a processing time of 2 minutes for a 512 by 512 image generated by SAH. It takes 6.74 seconds to reach the best tradeoff in applying SAED [15] with a 16×16 mask. However, neither Pang et al. nor Chang et al. provided timing details for each image or released which image they used either. Since we cannot find out their implementation details and the same hardware environment is hard to maintain, we do a rough timing comparison. Thanks to our new error diffusion method, our contrast-aware halftoning methods achieve faster speed. Our timing measurements are based on an Intel Core Duo CPU E8400@ 3.0GHz with 3GB RAM. When the mask size goes from 5×5 to 13×13, the processing time of our basic CED increases 5.3 times and the time of our Prioritized CED increases 0.8 times. The increase for our basic CED is around the same increase of the number of pixels in the mask while the increase of our Prioritized CED is much lower than the increase of the number of pixels. One explanation on this may be that the time of the basic CED that proceeded in $O(kN)$ is affected by k, related to the mask size and the time of the Prioritized CED that run in $O(NlogN)$ is invariant with the mask size. Thus, the increase of the processing time for our Prioritized CED due to the increase of the mask size is lower.

Parameters: Using a larger mask size usually creates higher quality halftoning in structure preservation while a smaller mask produces better tone matching. Figure 3.5 shows that mask size has a strong influence on quality. We believe that other halftoning algorithms encounter a similar problem of structure-tone tradeoff.

The parameter b in the weight calculation provides some control over the degree of tone matching and structure matching. Based on our experiments on over two hundred images, if b increases in some range around 0.5 to 4, the values for MSSIM and CPSNR are reduced, but the degree of tone similarity is increased. If b is greater than 4, our halftoning results are close to the quality of results from traditional error diffusion, which have very high tone matching but have very low matching in structure. The results miss many weak edges. If b is smaller than 0.5, the structure details are preserved well in our halftoning results but the tone matching decreased.

Color halftoning: Creating digital color halftoning is more complicated. To simplify, both the basic and Prioritized CED methods separately process each RGB channel to accomplish color halftoning. Figure 3.13 shows a comparison with SAH [139]. Our color results for halftoning show the structure more clearly. You can see the

(a) Original image

(b) SAH [139]

(c) Basic CED

(d) Prioritized CED

Figure 3.13: Color halftoning comparison.

right corner of the colored tree example in Figure 3.13. It is barely possible for us to see the individual leaves in the SAH result. Both our results shown in the bottom row gracefully express individual leaves and the distance levels are well arranged as well. Actually, color contrast is not a simple extension of intensity contrast. We expect that a more sophisticated color halftoning method would take into account interactions among colors.

3.6 Summary

In this chapter, we introduced contrast-aware methods with dynamic priority as a powerful and useful extension to classic error diffusion. Our halftoning results achieve very high structural matching due to contrast-aware weight distribution. The weight function attracts more positive error (or intensity) to light areas and increases the likelihood that darker pixels will become black to preserve local contrast. The dynamic priority-aware scheme assuring a distant pixel to be chosen as a candidate, in addition to the steeply dropping weights from the center, emulates the Poisson-Disc behavior that generates a good output of black pixels spatially. The introduction of priority scheme into halftoning is new and we demonstrated that it plays an important role in producing good results. Compared with other recent halftoning methods, we can conclude that the proposed contrast-aware halftoning methods have superior structure preservation. This work has been published as follows:

- Hua Li and David Mould. Contrast-aware halftoning. *Computer Graphics Forum*, 29(2):273–280, 2010

Prioritized CED has also been extended to generate other three styles: screening, stippling, and line art. The next chapter is going to demonstrate how to create screening with good structure preservation.

Chapter 4

Content-Sensitive Screening in Black and White

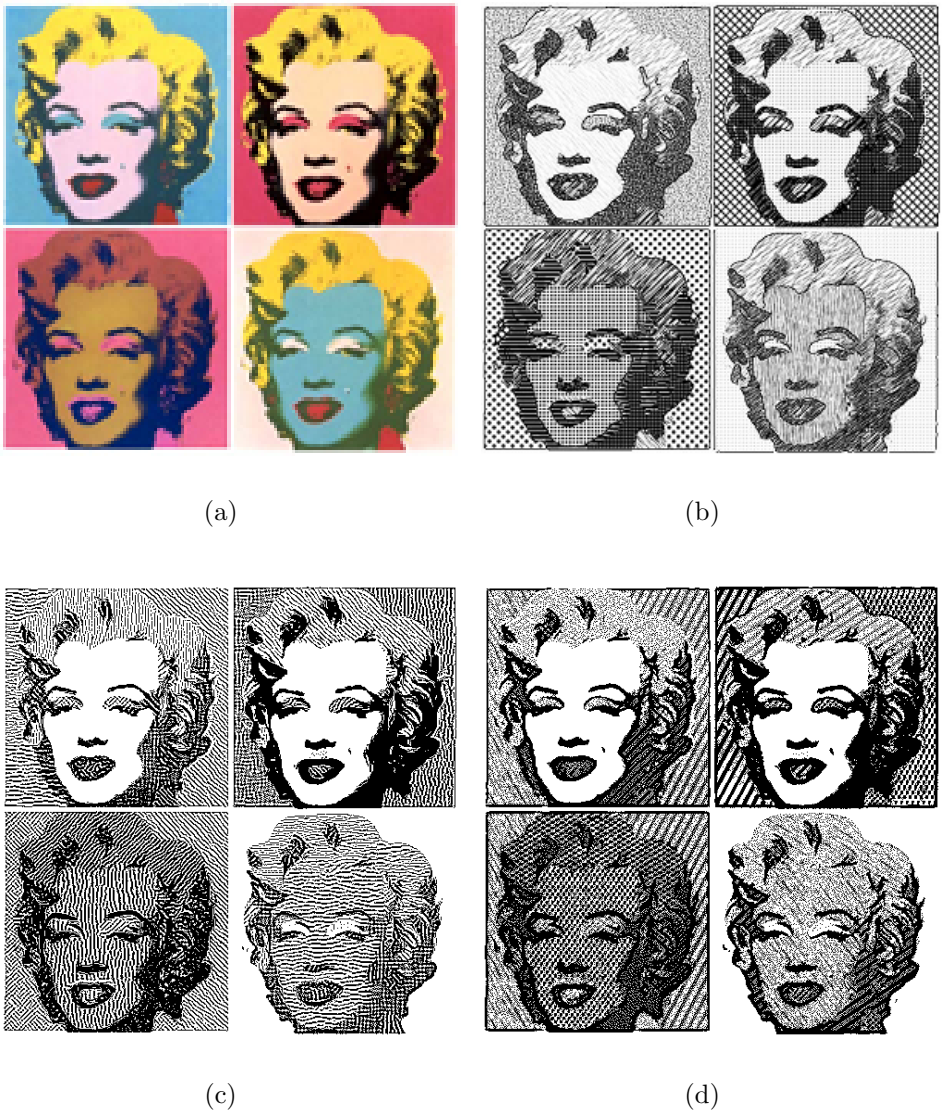


Figure 4.1: Marilyn Monroe art. Image resolution is 499 by 493. (a) Image reference; (b) richness-preserving Manga screening [142] (around a few minutes including segmentation); (c) and (d) our results using exclusion and textural patterns.

4.1 Introduction

The previous chapter described the Prioritized CED algorithm. The goal of this chapter is to explore additional capabilities of the framework for screening by introducing variations on priorities and mask properties. This modified Prioritized CED method can create content-sensitive screening (CSS).

A screening result is a white and black image with patterned effect. Ideally, a good screening algorithm should not only approximate the continuous tone of the original image and structural details, but also should have the ability to show different patterns. Dithering methods [174] create screening images by thresholding the image pixel values according to a predefined thresholding pattern. However, existing dithering methods solely pursued tone matching and did not consider structure matching at all. Screening results produced by dithering methods use unsatisfactory uniform patterns. Qu et al. [142] created high-quality screening for Manga, Japanese comic, with rich image content and various patterns. Their method for computer-generated Manga screening pursues tone matching as well as structural matching, such as texture. The authors reported that their screened Manga results are different from traditional screening because their screenings are not black and white, but greyscale. Qu et al. also mentioned that the Manga screening algorithm took a long processing time. To create screening with good tone and structure matching as well as a variety of patterns, we adapt the contrast-aware halftoning framework. We also introduce a content-based approach for pattern placement, which connects the pattern assignment with the textureness of a region. Patterns are generated by modifying the priority-based contrast-aware error diffusion in two ways. One variant uses a variety of exclusion-based masks to create exclusion patterns; the other variant is a multiple-stage pipeline to prioritize the processing order of external texture images.

An exclusion pattern is generated by an exclusion-based mask, which excludes a portion of pixels beneath a mask. The outcome of each error diffusion is to move error to the neighboring pixels, which lightens the nearby pixels in the dark area and darkens the nearby pixels in the light area. An exclusion-based mask has excluded and non-excluded pixels after diffusion. Excluded pixels are not affected by diffusion, which means they do not receive error and retain their intensity values. Priorities for excluded pixels are unchanged as well. Non-excluded pixels receive error and

generally reduce their priorities. As a result, the CSS system will likely process the excluded pixels earlier than the non-excluded pixels. For a dark area, since the excluded pixels are not lightened, they are more likely to be black than the non-excluded pixels. Therefore, there are more black pixels produced from the excluded pixels in a dark area. The exclusion direction in a mask is a direction for showing the patterns. Clustering more black pixels on the exclusion direction helps depict clear patterns. If the area is bright, the excluded pixels are more likely to be white. So black areas have more black pixels on exclusion direction and bright areas have more white pixels on exclusion direction. The increase difference of the number of black pixels between dark areas and bright areas brings a strong contrast image that preserves good structure.

We propose multistage screening process to create textural patterns that are obtained from the edges of texture images. The pixels in an image are grouped into a few sets. We process the sets in several stages. Each stage deals with one set of pixels. Priority-based scheme is used also. For each pixel, we apply contrast-aware error diffusion. The multistage screening system stops when all sets of pixels are processed. We prioritize the pixels with more interesting features in earlier stages and process the pixels with less interesting features in later stages. For example, a three-stage process can deal with the set of pixels on the edges of texture images first; the set of pixels on the ETF edges can be processed second; and we then deal with the remaining pixels last. This three-stage process can display textural patterns and the ETF edges. The use of Prioritized CED can produce good quality in structural preservation. The multiple-stage process allows us to introduce external patterns captured from image resources.

A comparison between our screening results and the result by the previous Manga screening method [142] is shown in Figure 4.1. The screened Marilyn's faces in (c) and (d) have varied patterns and structure richness, similar to the screening result in (b) from the Manga algorithm in terms of structure preservation. Our results even indicate the iconic mole on her face, which is totally lost in the Manga result. The Manga process has an interactive stage to ask users to select objects. It takes a few minutes for the Manga screening method to finish the process after the interaction, but our results can be automatically generated in less than a minute.

Contributions. We present two new ways, exclusion-based masks as well as a multiple-stage process, to generate non-uniform screening results in an automatic way. Our screening results naturally show the content without introducing segmentation artifacts unlike the existing Manga screening method [142], which displays artifacts from the segmentation boundaries. With no iteration for matching patterns, our screening process generally runs faster than the Manga algorithm. We also propose to assign patterns based on how much high gradients a region has. Our screening method can maintain structure details for a highly textured region by applying exclusion masks based on the edge direction of each pixel or the main direction of the region. This pattern assignment can preserve structural content well.

4.2 Screening Problem

The proposed screening method should find a way to cluster pixels to form patterns; and should present the difference between texture regions and non-texture regions by using patterns. We maintain the policy of the Prioritized CED method to offer the structure preservation in generating screening results. Forming a pattern requires clustering a set of black pixels to describe the shape of the pattern. The pixels on the shape of the pattern should be densely clustered black pixels and the pixels off the shape of the pattern should produce sparse or no black pixels. The difference in the density of black pixels indicates the pattern. Priority-based CED raises the difference in density of pixels for contrast awareness. We have to find a way to import the shapes of patterns into the scheme. We introduce exclusion masks and a multistage process to resolve the problem.

A screening result uses black pixels to show the patterns and important edges on a white background. We would also like to indicate original tone information in the screening result. Broken patterns occur when the areas are very bright. The system is going to output very few black pixels in the bright area, which are not enough to form a clear and complete pattern. The solution to avoiding broken patterns is to examine each region and prevent our system from assigning patterns in very bright areas, where broken patterns would occur.

4.3 Algorithm Overview

Figure 4.2 shows the outline of the CSS process. Given an original image, our CSS system can produce an image with screening effects. The system first segments the original image into regions using mean-shift [19] and a gradient map is calculated by applying the Sobel operators [166] to the original image. The system then starts to assign patterns to different segmented regions based on contents. This content-sensitive assignment is done by calculating how much textureness occupies each region. Highly textured regions should keep clear structures. Lowly textured regions are flat and assigning one type of pattern with tone matching would be enough. Patterns are generated in two ways: modifying the contrast-aware masks to exclude directional pixels, and adapting multiple stages to enhance textural patterns. Both ways are variants of contrast-aware halftoning. We design rules for assigning different patterns to highly textured regions and lowly textured regions. After finishing the assignments, our system produces rendered images with specific screening effects, which are exclusion patterns and textural patterns.

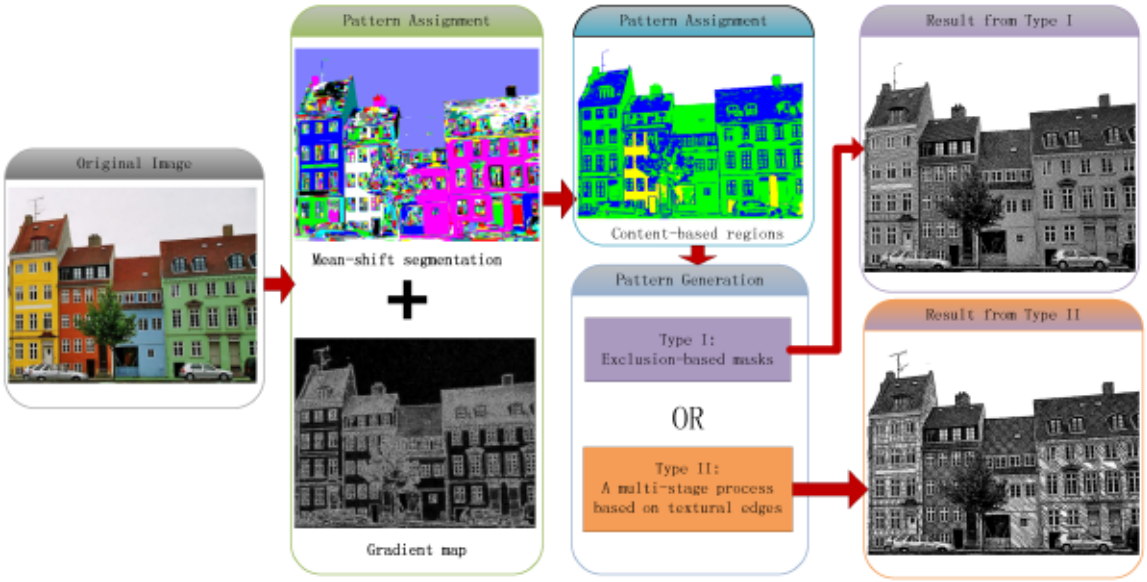


Figure 4.2: Overview of the screening procedure.

We use the mean shift method [19] for segmentation. Mean shift segments a region too much and produces many small regions. An object may be represented by a few segmented regions. The previous Manga screening system by Qu et al. [142]

used the mean-shift method to obtain segmentations too. They explicitly displayed segmentation boundaries in their output, which had distinct artifacts. Their screening results missed small details and showed many unwanted edges, visually not to be liked. The use of oversegmentation in our CSS will not have visible disadvantages in final screening for two reasons: we do not explicitly overlay the segmentation boundaries in our results, and we can preserve the structural details by our contrast awareness in the Prioritized CED. Segmented regions provide a guide to assign patterns. We propose a content-sensitive approach to assignment to help understanding the image, which further emphasizes structural details. Pattern generation creates two types of patterns: exclusion patterns and textural patterns. Exclusion patterns are created by using exclusion-based masks and textural patterns are generated by a multi-stage process. Additionally, the multi-stage strategy can give control over interesting edges by simply adjusting a few parameters.

As for the question about which pattern is used for a region, we do not treat it as a matching problem like the previous Manga screening method. Our content-based assignment helps the system determine the textured regions and the non-textured regions. The texture details in the textured regions are preserved by the aid of contrast-aware scheme, additional edge-exclusion masks, or the main edge direction. Other regions without textures are considered less important and the assignment is random. This strategy works for most images. However, the choice of pattern is not a trivial problem. For instance, two isolated regions might come from the same object and should use the same pattern. Without user intervention, in many cases there is no clue to find out if two isolated regions belong to the same object. The same problem occurs in the Manga screening algorithm.

4.4 Pattern Assignment

Mean shift produces a segmentation map. Applying the Sobel operators to the original image creates a gradient map. For each segmentation, pattern assignment is based on how much high gradients it has. Regions with high gradients have strong textures; and regions with low gradients look flat. Pattern assignment should preserve the orientation of textures in highly textured regions, whereas, in barely textured regions, we can have loose control, with no need to follow the edge directions. We can depict

clear object silhouettes by distinguishing regions with different patterns assigned in flat and textured regions. We calculate how much of each region R_i is occupied by texture. Regions with texture have high gradient values and regions barely with texture have low gradient values, calculated from magnitude values in the gradient map.

Each region has a statistical measurement H that is based on thresholding gradient magnitudes, computed as follows,

$$H_h = \frac{number(g(m,n) > T_h)}{\|R_i\|}, (m,n) \in R_i \tag{4.1}$$

$$H_l = \frac{number(g(m,n) > T_l)}{\|R_i\|}, (m,n) \in R_i \tag{4.2}$$

where the function $number(.)$ counts the number of pixels that satisfy the given predicate. The number of pixels in the ith region R_i of segmentation is written as $\|R_i\|$ and $g(m,n)$ is the gradient for the pixel at position (m,n). Equation 4.1 shows that H_h computes the proportion of the pixels with high gradient magnitudes in a region. In contrast, H_l calculates the percentage of the pixels with gradient magnitudes greater than the gradient threshold T_l. We write gradient thresholds as T_h and T_l with $T_h > T_l$. Since the intensities are in the range 0 to 255, gradient magnitudes are in a similar range. The examples shown in this thesis use $T_h = 100$ and $T_l = 35$. Users can try different values for both T_h and T_l. A higher value for T_h lowers H_h and similarly, higher T_l reduces H_l. The suggestion on $T_h = 100$ and $T_l = 35$ is based on testing a variety of images including photographs and illustrations. All results we have tested are good in structural preservation. We determine the assignment of patterns according to C_i, calculated by Equation 4.3.

$$C_i = \begin{cases} 0 : Empty, & \text{if } H_l < r_l \text{ and } \bar{I} > WH; \\ 1 : Low, & \text{if } H_l < r_l \text{ and } \bar{I} \leq WH; \\ 2 : Medium, & \text{if } H_l \geq r_l \text{ and } H_h > r_h; \\ 3 : High, & \text{otherwise.} \end{cases} \tag{4.3}$$

From the above, $\bar{I}$ is the average intensity value of the region; the thresholds for ratio of textureness are r_l and r_h; and WH is a threshold for a very light area. This thesis uses $r_l = 0.25$, $r_h = 0.6$, and $WH = 200$ to demonstrate results. In Equation 4.3, WH is an intensity value for a very bright area, $r_l = 0.25$ means

regions with little texture, and $r_h = 0.6$ are regions with intensive texture. Before users start the CSS method, they can specify the values for both r_h and r_l. The value of r_h, used in highly-textured regions, is supposed to be higher than r_l. The change of r_h and r_l will offer different degrees of adjustment on textureness. After the specification, the process determines assignment strategies based on C_i, a value to separate regions based on textureness. The separation helps us stylize the image content clearly. The values we suggest for r_l, r_h, and WH are the best values from the testing on many images in a wide range of contents. Rules to determine assignment are defined as follows.

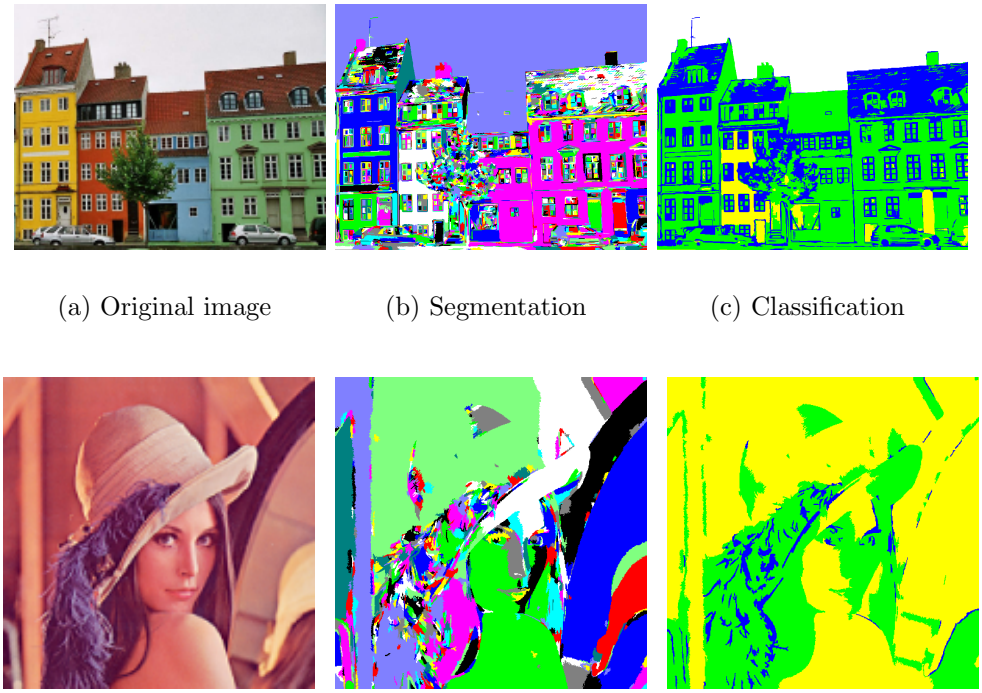
1. If C_i is zero for a region, this means the area is uniform and very bright, indicating a background or trivial area. Based on tone matching, a screened area in a very bright area produces very few black pixels. Broken patterns happen here, which can be distracting. A region with very little texture may have $C_i = 0$. Since the region barely has texture and is very light, we prevent broken patterns by leaving the area empty or using an arbitrary pattern to fill in.

2. If C_i is 1, the pixels in this region have the same and medium intensity values. We assign the patterns randomly. Since well-preserved structure and tone can be obtained by the contrast-aware strategy and the priority-based error diffusion, even random assignment creates good quality.

3. If C_i is 2, this region has medium degree of texture. We either represent the main direction of the region using patterns to display the texture or employ the edge-exclusion approach (described later) to match patterns with content.

4. For the remaining regions, C_i is equal to 3. These are highly textured regions and it is better to display all the information. We use basic contrast-aware halftoning without any variations, or adapt CAH with edge-exclusion masks (described later) in this case.

Figure 4.3 shows the segmentation and corresponding content-based classification according to this strategy for two images. In Figure 4.3 (c) and (f), blue regions are highly textured objects, such as the roof. Green regions show less textured regions, such as Lena's face and the building's body. Other yellow regions are uniform regions,



(a) Original image (b) Segmentation (c) Classification

(d) Original image (e) Segmentation (f) Classification

Figure 4.3: Content-based assignment.

without much texture. Background or trivial areas are colored in white, e.g., the sky area. We notice that the segmentation in Figure 4.3 (b) and (e) has not given a good partition for segmenting objects from the input image. The issues of oversegmentation due to the use of mean shift method are shown in the building's windows, roofs, and on Lena's face, but the segmentation can provide us a rough guide to help create patterns.

4.5 Pattern Generation

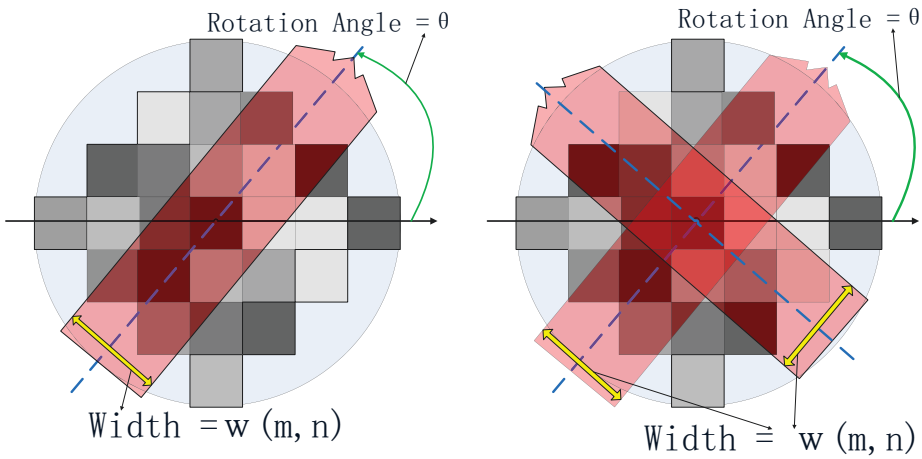
Next we propose two different approaches to generate a variety of patterns, including exclusion patterns and textural patterns. Exclusion patterns are generated by exclusion-based masks, which modify the priority order of error diffusion in an organized way. A multistage process preserves textural patterns captured in texture images. The following sections give the details.

Generation of Exclusion Patterns

We adapt Prioritized CED to create screening with non-uniform patterns. Exclusion-based masks are used, instead of the usual uniform circular masks. In the step of error diffusion, pixels in a mask absorb positive or negative error. The modified intensities of the neighboring pixels change their priorities, generally being lowered. We can look at a uniform region first. In this case, the priority values of all pixels in the region are the same before diffusion. After absorbing positive error, the neighboring pixels under the mask will be lightened; absorbing negative error makes the neighboring pixels darkened. Lightening happens in processing a dark uniform region and darkening occurs in screening a light uniform region. In both situations, error diffusion either increases the intensity values of dark pixels or reduces the intensity values of light pixels. Both actions of increasing and reducing move the intensity values toward the middle values in the range 0 to 255. The intensity value 128 has the lowest priority due to the maximum distance to the two extremes (black or white). Notice that both cases lower the priorities.

We propose to exclude specific subsets of pixels, not processing all pixels in a mask. The error will not propagate into the excluded pixels. The exclusion direction follows the direction of a pattern we design. The excluded pixels have unchanged intensities and the non-excluded pixels are lightened in a dark area or darkened in a bright area. For a dark area, more black pixels are clustered along the exclusion direction than the outcome without using exclusion masks. This way promotes the formation of interesting patterns.

Figure 4.4 shows two circular exclusion masks. Exclusion masks exclude pixels in one and two directions to create exclusion patterns. Each grid cell represents a pixel. The color of each grid cell is in greyscale whose value is proportional to the pixel's weight value, calculated by the contrast-aware weight distribution. The weight calculation is calculated according to Equation 3.4. The red strips in the figure show the excluded direction. The grid cells covered by the red strips are excluded pixels. Error diffusion will not affect the covered cells. The left mask has the pixels covered along one direction. As long as a grid cell is partially or completely covered by the red strip, the pixel in that grid cell will not be changed by the error diffusion. The angle θ and the $width$ represent the direction and the thickness of the exclusion line. The



(a) Exclude one direction

(b) Exclude two directions

Figure 4.4: Exclusion-based masks.

one-direction exclusion mask can represent patterns with only one strong direction, such as horizontal ($\theta = 0$) and vertical ($\theta = 90$) patterns. If the horizontal exclusion is chosen, pixels along the horizontal direction are excluded. Those excluded pixels have unchanged intensities after diffusion, and their priorities are thus the same. Other non-excluded pixels have their intensities changed after diffusion, generally, lowering their priorities. Each error diffusion in dark regions will not lighten horizontal pixels; hence those pixels have their darkness unchanged. Horizontal exclusion masks can produce more black pixels, clustered horizontally, than normal circular masks. In contrast, light regions produce more white pixels horizontally. The increase of difference helps preserve contrast.

We can control exclusion direction, for example, aligning it with the edge direction. A mask excluding pixels along edge direction is called an edge exclusion mask. The direction follows the edge tangent from the center pixel during the error diffusion. When edge-exclusion masks are applied to an image, exclusion angles are changed based on current edge direction pixel by pixel. There will be more black pixels clustered for displaying the edges in black regions than the black pixels produced without using edge exclusion masks. Since the edge exclusion follows the directions of edges, patterns formed by black pixels in dark regions follow the edges too. Clustering

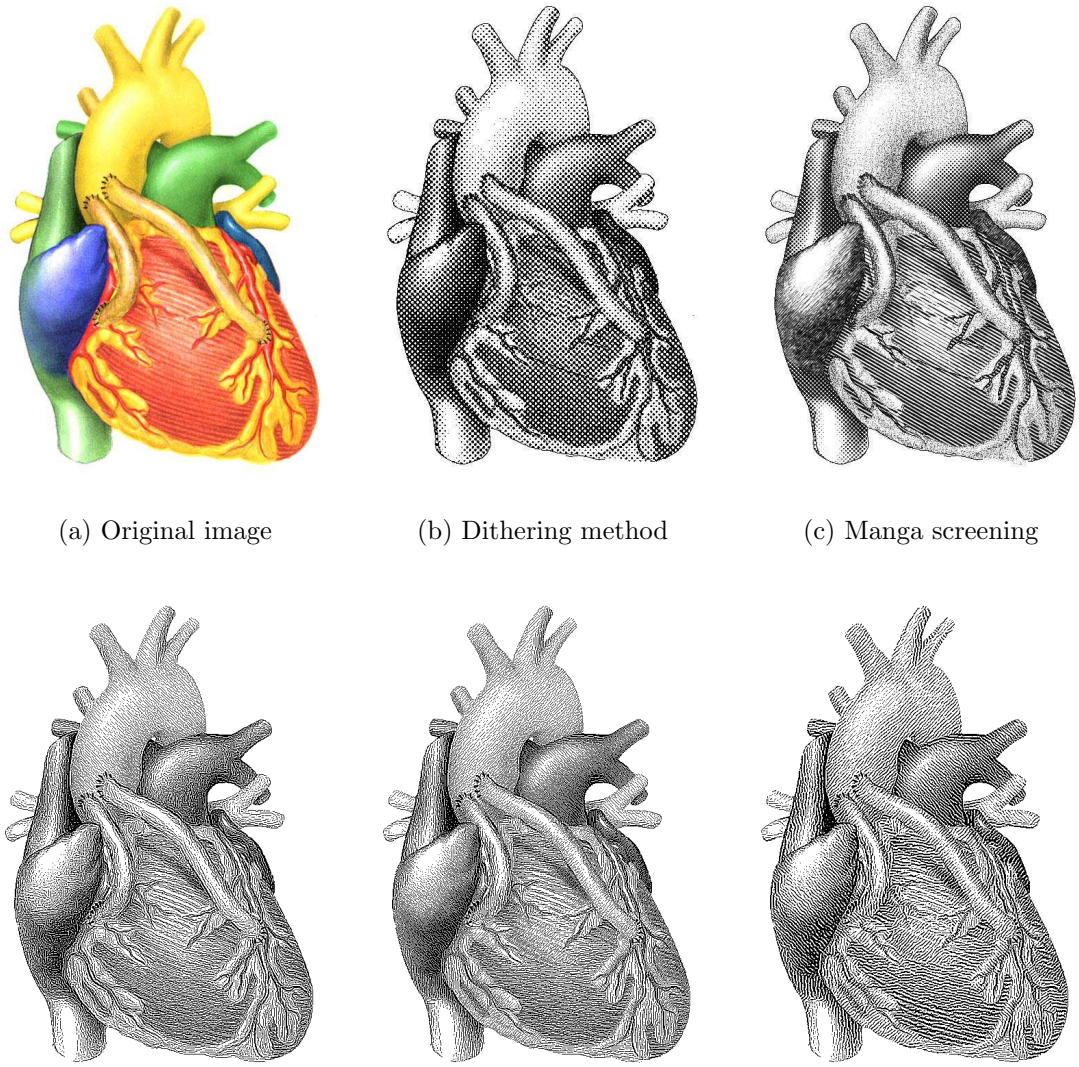
more black pixels at the edges can promote structure details and textures. Edge-oriented weight distribution in edge exclusion masks enhances structures; we use the weights for regions with $C_i = 3$ that are highly textured regions or sometimes for regions with $C_i = 2$ that are averagely textured regions. The choices on $C_i = 2$ and $C_i = 3$ are decided by users before the algorithm starts. Beyond edge exclusion masks, another way for regions with $C_i = 2$ is to use the main direction calculated based on the region to guide the angle θ. Notice that there is one big difference between edge-based exclusion and the use of the main direction. If a region is processed with edge-based exclusion, each pixel of the region changes its exclusion angle based on the edge direction. Conversely, the main direction of a region is constant for all pixels in that segment, and therefore the exclusion angles for all masks are the same within the region. Figure 4.4 (b) shows an exclusion-based mark that excludes two directions, along with parameters θ and $width$, similar to the way of one-direction exclusion. The use of two exclusion directions can produce crossed patterns.

Some results using exclusion-based masks on a heart image are shown in Figure 4.5. We summarize the decision on assignment as follows: if C_i is 0, we leave the regions empty (blank); if C_i is 1, we randomly choose the patterns for the exclusion angle; otherwise, we use edge-exclusion masks or use the main direction of each region, chosen by the user specification. Figure 4.5 (b) was a screening that was generated by Qu et al. [142]. Our results have non-uniform patterns and look much better than the result from dithering. We compare with the result from Manga screening, shown in grayscale in Figure 4.5 (c). Our screening shows the small veins very clearly, even in black and white.

We provide adjustment to the thickness of the exclusion width to add variations on visual appearance. Thickness $width(m, n)$ is calculated as follows:

$$width(m, n) = T_{max} \times (1 - I(m, n)/255) \tag{4.4}$$

where T_{max} is the maximum thickness ($T_{max} = 6$ here). A high value of T_{max} provides very thick patterns and a low value produces thin patterns. We know that T_{max} must be less than the size of masks. If T_{max} is greater than the mask size, the exclusion mask excludes too many pixels. Figure 4.5 (f) shows an example of thickness control. Two different screened images appear in Figure 4.6, showing the thickness control for different patterns. Both screenings clearly display structure details with enhanced



(a) Original image

(b) Dithering method

(c) Manga screening

(d) $C_i = 2$ with edge-exclusion masks

(e) $C_i = 2$ with main direction

(f) $C_i = 2$ with thickness control

Figure 4.5: Screening comparisons.

pattern effects. The patterns look thicker and clearer. The left image in Figure 4.6 uses four patterns. The four patterns are randomly selected for the uniform regions, which have $C_i = 1$. Uniform regions do not have many structural details and the random patterns will not cause any loss of content. The right image in Figure 4.6 uses six patterns. The six patterns are randomly selected for the uniform regions. All patterns are defined by exclusion-based methods.

Edge-exclusion masks and the main direction: When a region is processed,

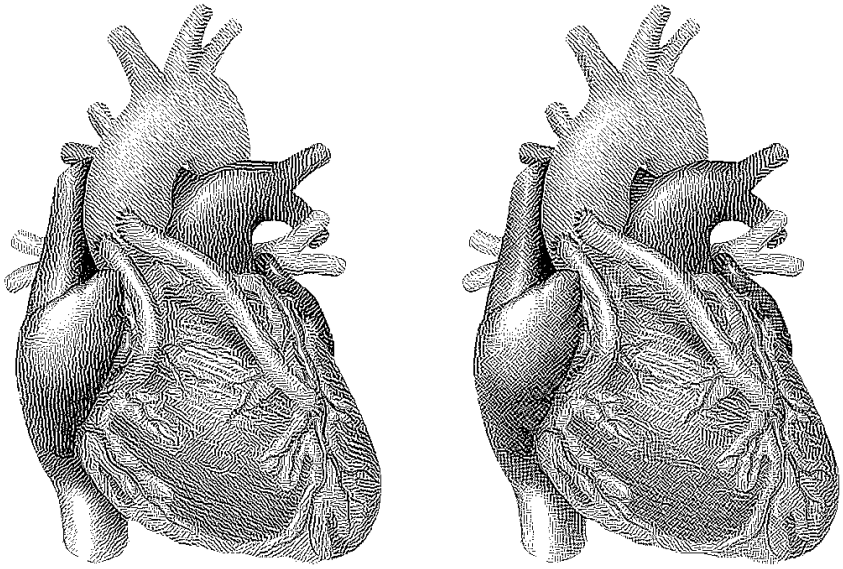
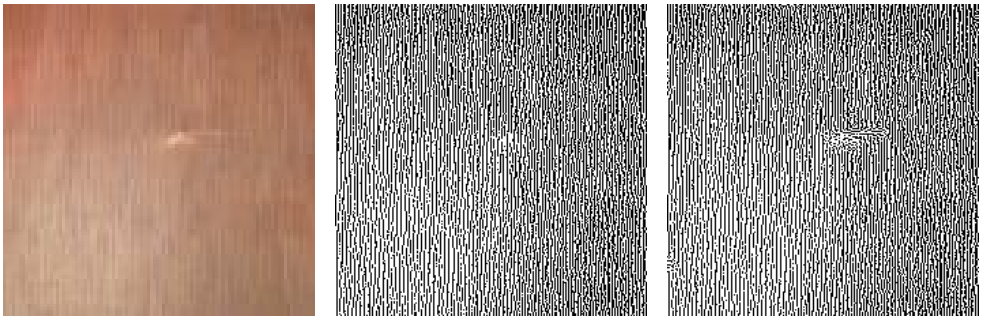


Figure 4.6: Thickness control. Left: 4 patterns; right: 6 patterns.



(a) (b) (c)

Figure 4.7: Two ways. (a) Original texture; (b) main direction; (c) edge exclusion.

edge-exclusion masks modify their exclusion angles pixel by pixel; in contrast, the use of the main direction in a region maintains the same exclusion angle for all pixels. The results from edge exclusion in general are more appealing than those from the method using the main direction. Figure 4.5 shows the difference between exclusion-based masks and main directions. Visually, both are quite good at retaining structural details. In Figure 4.7, a textured image is processed by both main directions and edge-exclusion masks. The edge-based exclusion not only shows the main trend of this texture, but also grasps the small dent, barely present in the result from the

(a) Without

(b) With edge exclusion

(c) Main direction

(d) Edge exclusion plus thickness control

Figure 4.8: Content-sensitive assignment.

main direction. The small dent is only in a vertical direction in Figure 4.7 (b); while the dent area in Figure 4.7 (c) has black pixels not only in a vertical direction, but also in a horizontal direction. Another example in Figure 4.8 shows the different effects from both types of direction control. We include a screening result processed without using our content-sensitive assignment in (a). As you can see, with edge-exclusion masks in Figure 4.8 (b) and Figure 4.8 (d), the hair, eyes and the nose look much better than Figure 4.8 (a). Figure 4.8 (c) is a screening result using the

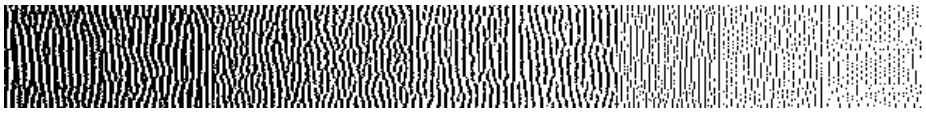
main direction. Both exclusion-based modification and the main direction match Lena better than Figure 4.8 (a). For tone we show a ramp in Figure 4.9 for five

(a) Original ramp

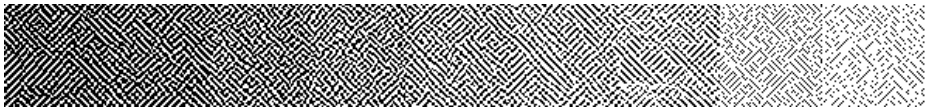
(b) One direction, $\theta = 45°$

(c) One direction, $\theta = 135°$

(d) One direction, $\theta = 0°$



(e) One direction, $\theta = 90°$



(f) Two directions, $\theta = 45°$

Figure 4.9: Ramp with different patterns.

exclusion-based patterns. Visually, all patterns are reasonably able to convey tone differences.

Textural Patterns by a Multi-stage Process

The priority-based scheme is flexible to provide promotion of interesting edges. We propose a variant of our priority scheme. The algorithm separates an image (S) into a few pixel sets $S_1, S_2, \ldots, S_N$ and processes them in several stages. The first-stage priority set (S_1) will be processed first; the pixels belonging to the second-stage priority set (S_2) will be dealt with second; and so on. After the system finishes all priority sets, in a final stage, it processes any remaining pixels (S_N).

We would like to maintain a pixel's initial tendency of intensity from the original image. A dark pixel would retain its tendency to black and a bright pixel would retain its tendency to white. In this way, we preserve the contrast. The purpose of error diffusion is to compensate the loss of tone in the way that dark regions diffuse positive error to lighten pixels and light regions diffuse negative error to darken pixels. Both lightening in dark regions and darkening in bright regions violate pixels' initial tendency. A pixel in a later stage would have received error a few times from the earlier stages. The more times a pixel has received error, the less the pixel could maintain its initial tendency. Pixels processed in earlier stages have less opportunity to be affected by error diffusion than pixels in later stages, since once a pixel has been processed, it no longer receives error. This set of pixels in earlier stages are more likely to maintain their initial tendency. For a dark area, more pixels keeping their initial tendencies could produce more black pixels. If we would like to output densely clustered black pixels on edges in dark regions, we should process edges earlier. Thus, we assign the pixels on important features, such as edges, to an earlier stage to make them more likely to maintain their original tendency. The remaining pixels, considered as less important content, are processed later. This is a multi-stage process.

We can generate the priority set in two ways: it could be created manually; or, it could be generated automatically, using low-level features such as edges or high-level features such as faces. The minor drawback is that the outcome is dependent on the quality of the priority field. Our multi-stage process has the advantage of allowing us to use the framework effectively and flexibly. One possibility is to define a set of pixels by the edges from texture images. The pixels on the edges will be sent to an earlier stage first, which promotes the edges. The generation of textural pattern uses texture images. See Figure 4.10, which shows a three-stage process.

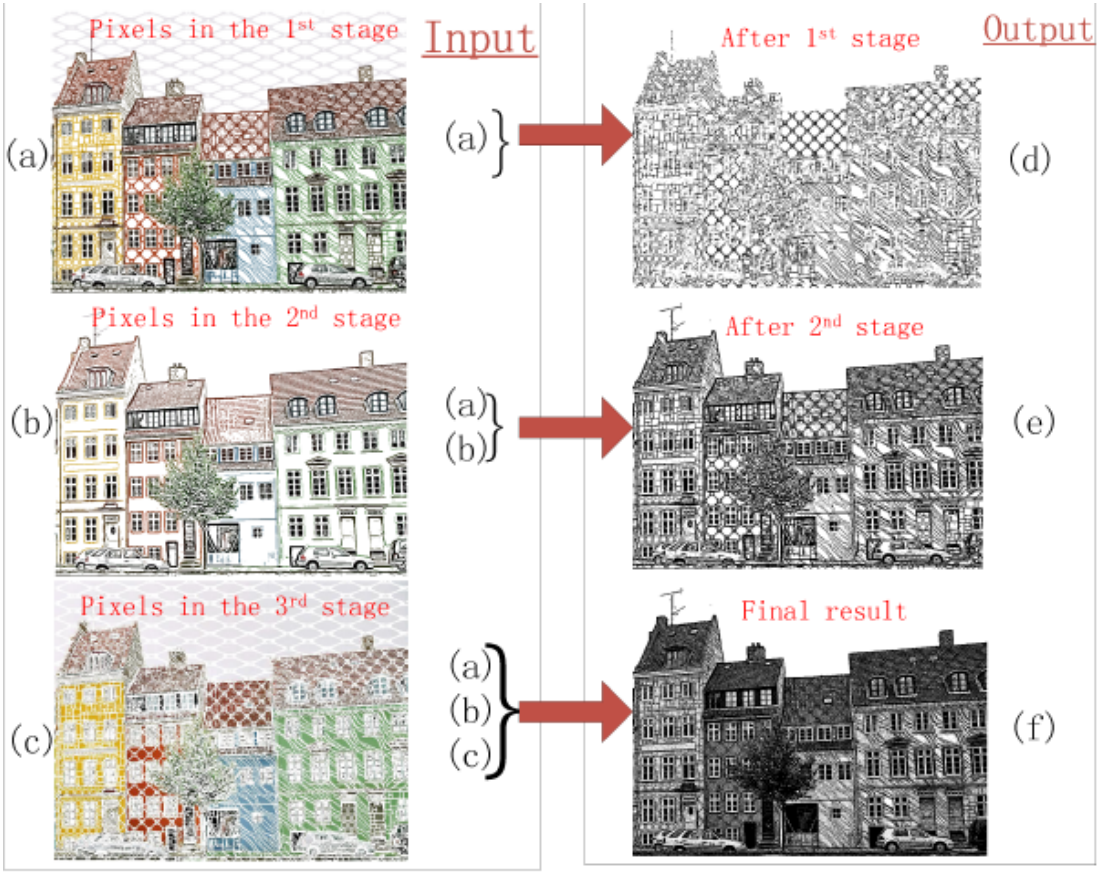
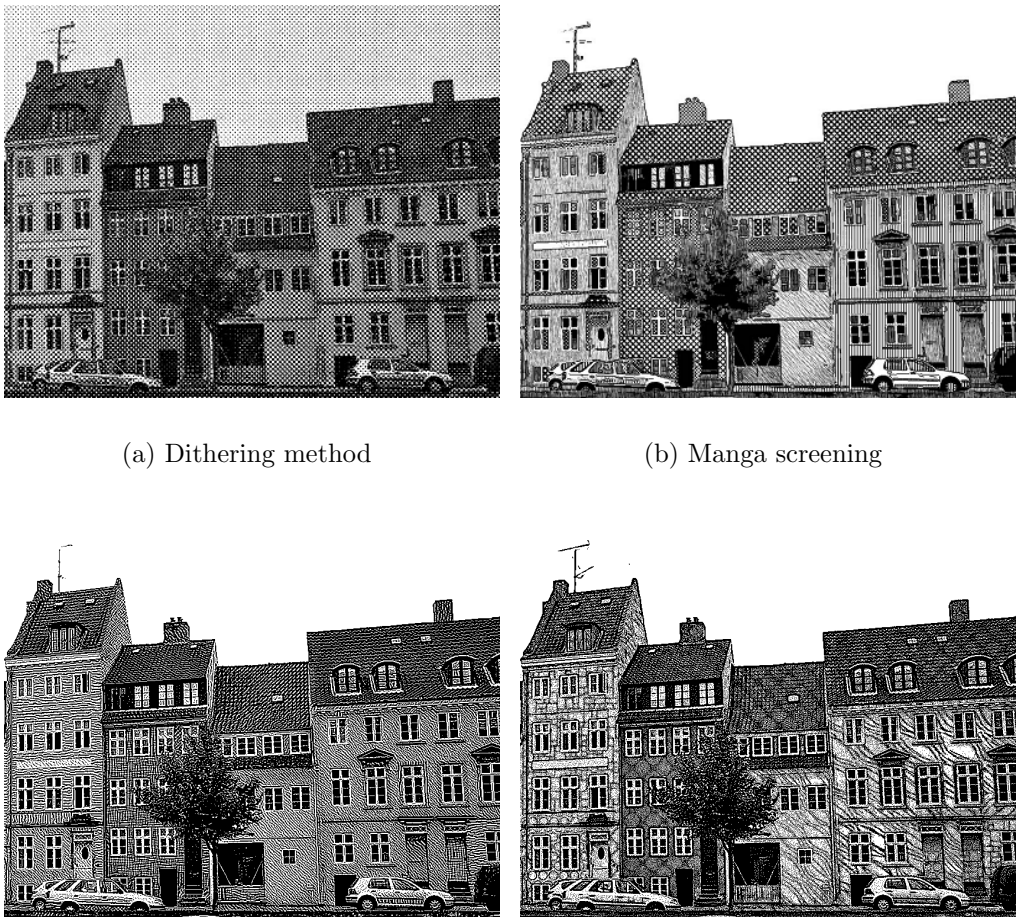


Figure 4.10: A three-stage process for promoting both textures and ETF edges.

In Figure 4.10 (a), the colored pixels show the pixels to be processed in the first stage, which are the textural edges. In Figure 4.10 (b), the second stage deals with pixels on the edges, obtained from thresholding the Edge Tangent Flow (ETF) [71]. Finally, the last stage processes the remaining pixels. Since we are interested in presenting the edges from textural patterns, those pixels at textural edges are set earlier. Through this configuration, the final screening can easily enhance the patterns from input textures. The requirement of tone and structure in screening can be satisfied from error diffusion also. Because we could use any texture, it is possible for textural patterns to have a visual appearance different from an original image. A worse case could be that the tone may be opposite to the original tone. An improvement by matching the tone and the structure with those of the input patterns could refine the assignment. The goal of the improvement is to see if the patterns are suitable for that region. However, the multi-stage process does not have the matching



(a) Dithering method

(b) Manga screening

(c) Exclusion screening

(d) Textural screening

Figure 4.11: Screened building comparison.

stage in the procedure, the content-sensitive strategy in the multistage process plays a role, not as important as that in exclusion pattern generation.

Figure 4.11 presents a screened building processed by different screening methods. The dithering result looks less appealing than other results because the pattern is always the same, and there is no structural preservation. Manga screening in greyscale looks nice, but is not a binary reproduction. Both exclusion and textural screening results use different patterns to distinguish regions and preserve many fine details for this image. Look at the cars and the tree: they have no segmentation artifacts while the Manga result has. Compared with the exclusion screening, the textural screening can have more varied texture patterns, which provide diverse effects for non-uniform screening.

Image	Size	Mean Shift (s)	CSS (only) (s)	Total (s)
Figure 4.1 (c)	499× 493	21.74	11.65	33.39
Figure 4.1 (d)	499× 493	21.32	12.3	33.62
Figure 4.5 (d)	522×689	29.3	10.97	40.27
Figure 4.5 (f)	522×689	29.6	10.24	39.84
Figure 4.11 (c)	839×663	40.84	29.29	70.13
Figure 4.11 (d)	839×663	40.47	27.76	68.23

Table 4.1: Processing time on an Intel(R) Core(TM) i3-2350M CPU @2.3GHz with 8GB RAM .

4.6 Results and Discussion

After parameters are set up, content-sensitive screening can automatically generate screening effects. The proposed screening method is a variant of error diffusion based on contrast and priority by applying either exclusion-based masks or a multiple-stage process. The consideration of contrast during the error diffusion helps CSS method preserve the structural details. A further improvement of visual appearance is from the content-dependent assignment.

We cannot provide a thorough comparison with previous Manga screening [142] in terms of timing details since the authors did not report the processing times for all images. Qu et al. did report the hardware they used was a PC with P4 3.2GHz CPU, 2GB memory. However, we cannot know the implementation details and Manga screening needs manual intervention to regroup small regions to avoid fragile output because of oversegmentation. They mentioned that the interactive editing for users was very quick but we do not know the time detail. Qu et al. mentioned that processing an image of size 800×1000 needed four minutes. We provide a rough comparison based on an Intel Core Duo CPU E8400@ 3.0GHz with 3GB RAM. Processing 800×1000 image takes 72 seconds for the segmentation and takes 110 seconds to obtain our screening result in total. If we exclude the segmentation time, our process only takes 38 seconds. Based on this timing, we probably can provide users an interactive tool for generating screenings. Table 4.1 provides more information about processing time for our other results.

We compare our screening results with the Manga screenings by Qu et al. [142] in a side by side way. Manga screenings are greyscale and our screening results are in black and white. Although side by side comparisons are not fair for both of us, they at

least show the visual difference from both methods. Mean-shift segmentation for constructing small regions for objects is used in both Manga algorithm and our screening method. Manga screening results show the segmentation boundaries explicitly, which are not natural visually. The many small regions cannot destroy the structural quality in our screening results, which present the object boundaries naturally because we did not explicitly overlay the boundaries on the top of the screening.

The assignment is not dealt with as a colorization problem, as Manga screening did. Our proposed system runs in a single process, which follows the intrinsic tone and content that guide the patterns and the assignment. The images shown in this thesis are the same images used for the Manga paper [142]. From this choice, we can use side by side comparisons to have a visual comparison with Manga screening. We also have tested our screening algorithm on a variety of images, including photos and illustrations. The image content is in a wide range including natural scene, people, objects such as cars, and buildings. All screening images from our CSS algorithm have very good quality in structure preservation. Our algorithm failed when a few regions belonged to the same object and were not assigned the same pattern. This is because the mean-shift method oversegments the object and we have no way automatically to find those regions belonging to the object. This situation happened in Manga screening too. As Qu et al. did, we could ask users to select the objects interactively and allow them to explicitly tell our CSS system to assign the same patterns for all regions belonging to the same object.

The multi-stage process has flexibilities to show different types of importance. Figure 4.11 and 4.12 show edge-tangent flow (ETF) edges combined with different textural edges. The ETF edges are promoted without introducing artifacts or false edges, which are common problems occurring in some previous screening methods [13, 169].

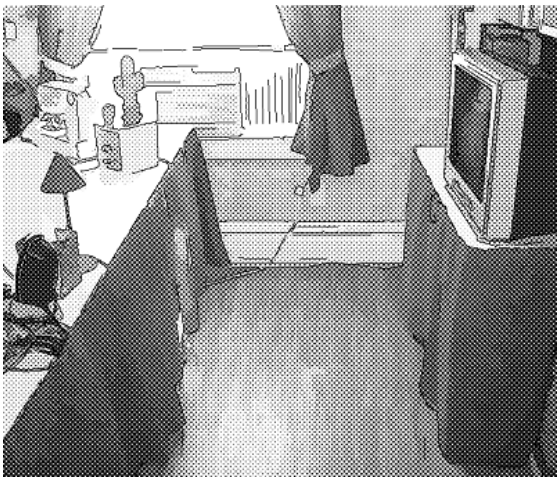
Our scheme is open to be extended to color screening by processing three RGB channels individually. Figure 4.12 shows two colored screening examples with textural patterns. Both examples display interesting and distinct effects, though not as yet fully explored. However, enough content details are shown to convey the main information of the image.

Figure 4.13 shows a comparison between content-sensitive screening and previous

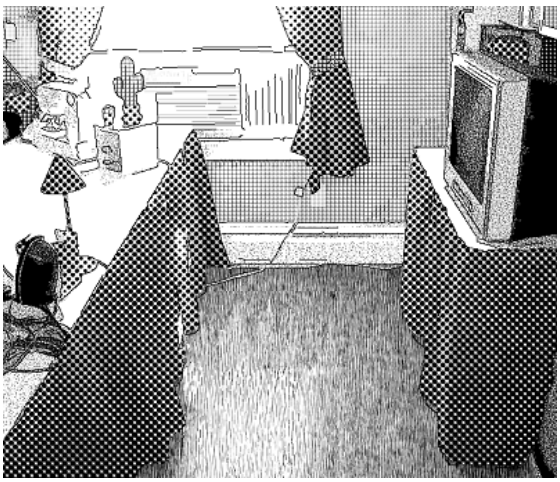
Figure 4.12: Color screening.

screenings. It is certain that our screening looks much better than the result by the dithering method. Manga screening looks reasonable, but places the segmentation boundaries on top of its patterns to distinguish between objects. Notice that the edges between cloth and the desktop have annoying boundaries of objects on the desk. The unnatural boundaries are artifacts from segmentation. Our screening result has no segmentation artifacts. The structural details are well-preserved by contrast-aware masks, the priority-based scheme, and the content-sensitive assignment. Exclusion masks using edge directions or the main direction of a region can further enhance structural contents. However, there is a pattern assignment error because we used

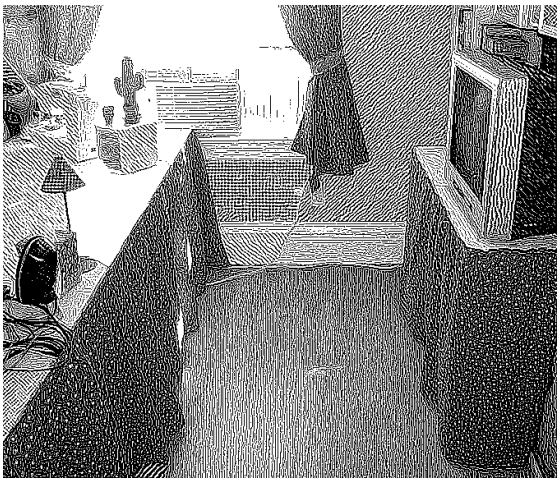
(a) Original image



(b) Dithering method



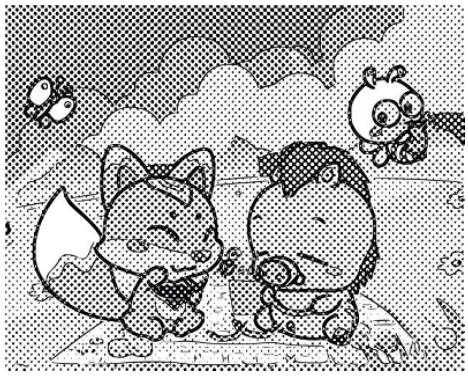
(c) Manga screening



(d) Exclusion patterns (thick)

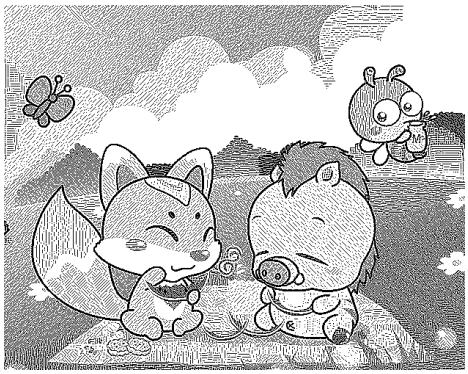
Figure 4.13: Another comparison of screening results.

mean-shift segmentation. We did not check if two regions belong to the same object. We might assign two different patterns to the same object. For example, the wall in Figure 4.13 is supposed to be in a single region. The segmentation method separates the wall into two parts because there is a wire crossing. Based on the tone, we have no way to tell our system those two parts are the same wall. It seems that this situation should use user intervention to fix up by manually assigning the same patterns to both regions. Manga screening faced the same problem as well.

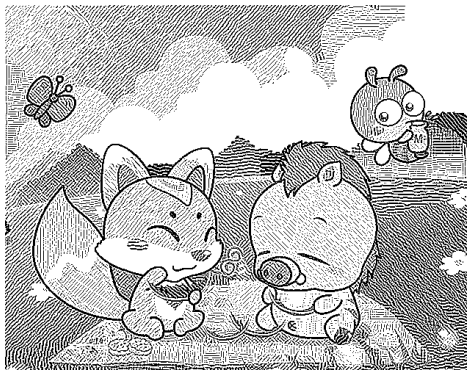


(a) Dithering method

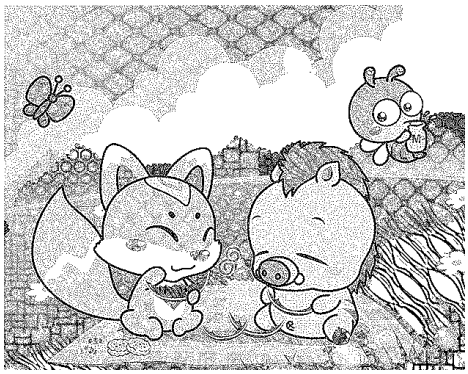
(b) Manga screening



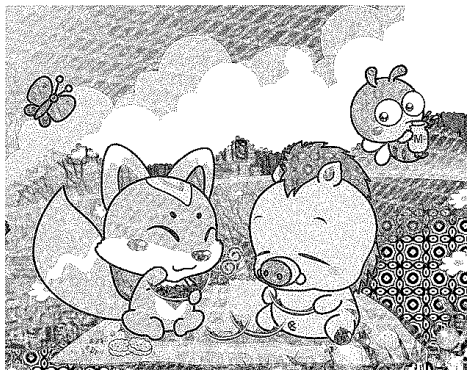
(c) Exclusion screening



(d) Exclusion screening (thick)



(e) Textural screening



(f) Textural screening

Figure 4.14: Comparisons of screening results.

Another comparison shown in Figure 4.14 illustrates a cartoon screened by different methods. Our results generally look better than the dithering result and the

Manga screening. One continuous object surface is supposed to use the same patterns to fill in. Sometimes region differences are overemphasized. Because the assignment process has no way to determine the semantic meaning of each region, the assignment may mistakenly use more than one pattern to fill the space.

The parameter values we used in the CSS system have been tested on a variety of images including natural scenes, illustrations, cartoons, and portraits. We always obtained good results. Our tests showed that the system is robust. Users can specify the parameter values before the process starts. Different parameter values may change tone or structural quality in an acceptable way. The only issue here is the assignment error, such as the wall patterns in our result shown in Figure 4.13. A possible solution to this assignment error is to correct the error by specifying the patterns for regions manually.

4.7 Summary

In this chapter, we demonstrated how screening can be generated by adapting the Prioritized CED technique. The main focus was on showing different screening patterns in black and white with good structural details generated automatically. The ideas for exclusion-based masks and the multistage process can be adapted to both image processing and non-photorealistic rendering, such as filters used for edge detection. The next chapter is going to revisit those two variations in the context of stippling. The content-based assignment for patterns avoids the matching problem and saves computation time. However, our assignment fails when an object surface is separated by other objects. The individual parts belonging to the same object may need user intervention to tell our algorithm that the parts should be assigned the same patterns. Compared with existing dithering methods and other structure-aware screening such as Manga screening, our results nicely show many structure details and have a variety of patterns to select. Our screening method is also competitive in speed. However, an approach to matching a best-fitting pattern to a region could improve the visual appearance of a screening for sure.

This work applied for screening has been published as follows:

- Hua Li and David Mould. Content-sensitive screening in black and white. In *International Conference on Computer Graphics Theory and Applications*

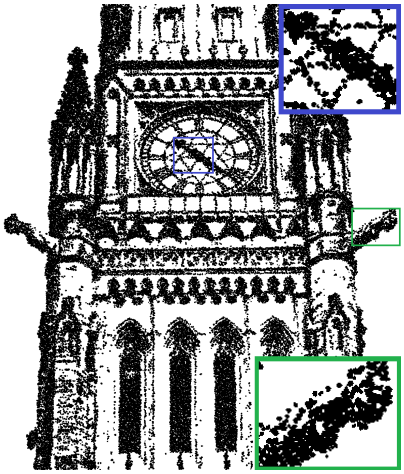
(GRAPP), pages 166–172. SciTePress, 2011

Until now, contrast-aware consideration with dynamic priority has been used to obtain a distribution of black pixels for both halftoning and screening effects. The idea works well for enhancing important features. The next chapter is going to demonstrate how the Prioritized CED framework can be adapted for stippling generation.

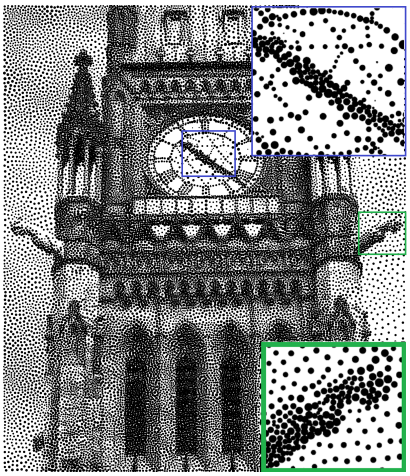
Chapter 5

Structure-Preserving Stippling

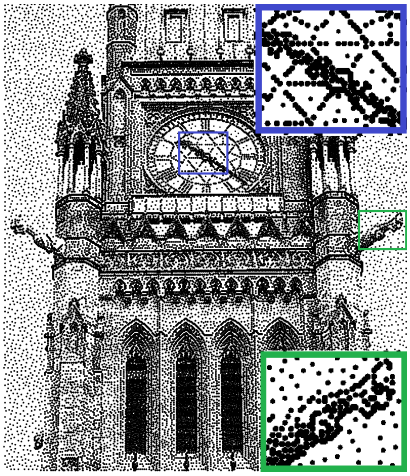
(a) Original image



(b) Martín et al.'s method



(c) Secord's method



(d) SPS

Figure 5.1: Stippling comparisons. Two zoomed-in views of the clock face and the sculpture are shown. (a) Original image; (b) Example-based stippling method applied to manually enhanced image [117]; courtesy of Domingo Martín; (c) Secord's stippling method applied to histogram-equalized image; 30,209 stipples by Secord's method [161]; (d) 29,574 stipples with $P(5, 3, 0, 7)$ by SPS method.

5.1 Introduction

The previous two chapters introduced the Prioritized CED algorithm and demonstrated that our algorithm worked successfully for generating halftoning and screening. The goal of this chapter is to describe how this algorithm can be adapted to stippling generation, and still retain the advantages of high quality in structure preservation and good performance. We call our proposed method Structure-Preserving Stippling (SPS). We aim to achieve two tasks: (1) providing a reduction of the number of dots in stippling; (2) exploring applications of stippling in different styles.

We generated halftoned and screened by using the techniques: contrast-aware weight calculation, dynamic priority, mask variations, and a multiple-stage process. The proposed framework for SPS also employs those techniques. In addition, our SPS method introduces a new modification, a nonlinear spatial function to shrink or exaggerate errors. The nonlinear function implicitly provides global adjustment of the number of dots. Simultaneously, the adjustment controls the coefficients of weights during error diffusion in a contrast-aware way. The stippling can preserve structure even with very low stipple budgets. We explore a unified framework of stippling to create a variety of artistic effects, including heightening, pointillism, painterly, screening with dots, and scratchboard.

Figure 5.1 shows a comparison with two previous stippling methods: Martín et al.'s [117] and Secord's [161] stippling methods. Both stippled results by Martín et al.'s and Secord's methods are generated after preprocessing the original image to improve structural quality. Martín et al. used a software to manually darken and lighten image content in order to strengthen image contrast. After manual image enhancement, they employed an example-based method to generate stippled images. Figure 5.1 (c) was generated by applying Secords method to an image pre-processed by histogram equalization. Our SPS method used the original image without any preprocessing. Our stippling result shows the key details very clearly; for example, look at the zoomed-in windows for the clock hands and the arch, which Martín et al.'s and Secord's methods cannot depict very well.

Our SPS method provides the density control of the stippling. We investigate variants of our SPS method for generating different effects. Figure 5.2 shows a variety of styles including scratchboard, high-level stylization, dot-based screening, and

pointillism, all of which are based on our stippling framework.

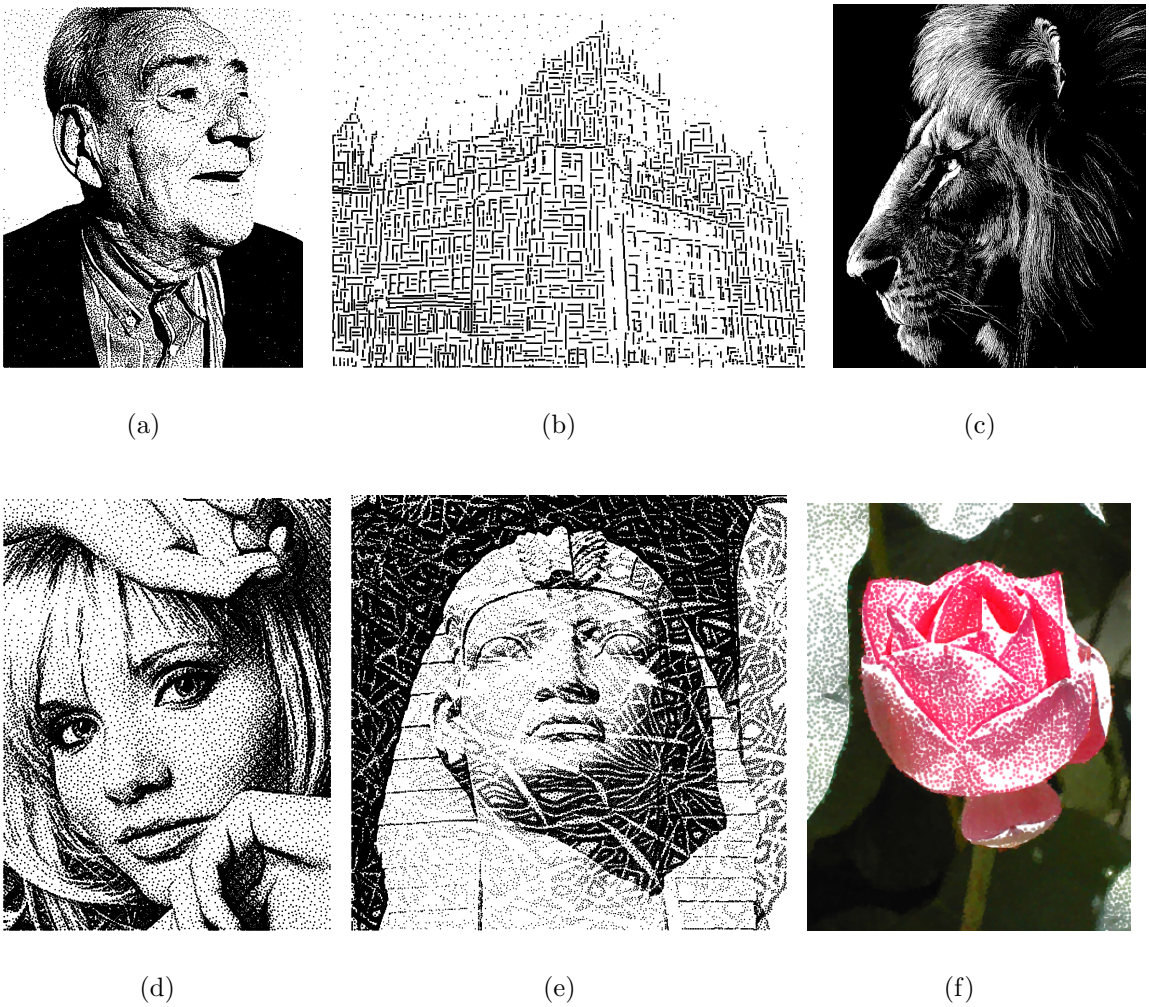


Figure 5.2: Stylized stipplings using SPS with dots. (a) SPS; (b) high-level stylization; (c) scratchboard; (d) screening; (e) patterns; (f) pointillism.

Contributions. Our SPS method is fast and automatic. SPS can create stippling with diverse new effects, which previous researchers did not investigate. The proposed structure-aware stippling mechanism can reduce the huge number of pixels used by contrast-aware halftoning to a relatively small number of stipples while preserving structure. This advantage of structure preservation over many stippling methods lends SPS to evolution into many artistic styles including heightening, pointillism, painterly, screening with dots, and scratchboard. The different effects and styles are built on the same single framework, which is not easy to achieve.

We propose to control the number of dots with a nonlinear function that shrinks or exaggerates errors, depending on the size of dots. We also propose new weight distribution schemes for different stippling styles by using irregular masks and line masks. Irregular masks are produced by an Eden-like growth scheme [29] and can create irregular stipple distributions. Line masks employ line-shaped masks and can present strange but interesting texture effects such as rough surface. A set of variations on colors are introduced to bring a series of artistic styles, all of which visually look appealing due to the high quality of structure preservation.

5.2 Stippling

To generate stippling, we have to consider the size, the shape, and the location of each stipple. One way to represent strong contrast in a stipple distribution is to enhance the difference in stipple density. Our proposed basic structure-preserving stippling method extends contrast-aware halftoning by adding the adjustment of the number of stipples. The adjustment depends on the size of dots, the parameters used in CAH generation, and the new nonlinear function for shrinking and exaggeration.

Existing computer-generated stippling methods seek a regular distribution, evenly placing dots in space. However, artists' work has irregularity due to imprecise placement or deliberate preference in their stippled works. In this chapter, we not only seek structural preservation, but also focus on new possibilities to create a variety of stylistic capacities by presenting different non-traditional distributions of dots. Our stippling method can create various styles including heightening, scratchboard, and pointillism. The variations include modifying colors, masks, and priority configurations.

5.3 Structure-Preserving Stippling

In this section, we describe the details of the basic structure-preserving stippling method. The basic SPS method extends Prioritized CED by using a non-linear function to adjust the error, which provides control over the density of stipples. To make up for the loss of tone when the number of stipples is low, we propose stipple resizing to improve tone matching.

5.3.1 Basic Structure-Preserving Stippling

To adapt a stippling algorithm from an error diffusion algorithm, the system employs black dots rather than black and white pixels. Naïvely, it could simply replace every pixel with a black stipple, but we cannot afford the extremely high stipple count and the resulting image would have many overlapping stipples, which would not resemble a stippled image at all. Error diffusion algorithm has a step to decide if a pixel is black or white. The decision on the placement of a pixel guides us to place a stipple. We can process the pixels in a prioritized order, draw a stipple when the intensity value of this pixel is below the threshold, and then diffuse the error. The outcome of processing a pixel is a decision whether or not to place a dot. If the outcome at the location is to place a stipple, the difference between the original intensity and the black output is positive. Placing a stipple makes the location become darker than its actual value, so to compensate the darkening we need to lighten nearby pixels. Conversely, if the decision is not to place a stipple, the process incurs negative error – the location becomes lighter than its actual value, so nearby pixels are darkened. After positive or negative error is distributed to future pixels, the intensities of unprocessed pixels are changed and this change would affect the decision about whether to place or not to place a stipple. We can adjust the distribution of error to give control over the density of stipples, since the error affects the number of stipples to be created.

Deussen et al. [24] mentioned that the size of stipples in real stippled works rarely varies by more than a factor of two. We follow Deussen et al.'s suggestion and use the maximum size of stipples twice the size of the minimum, so $r_{\text{min}} = 1$ and $r_{\text{max}} = 2$ in Equation 5.1.

$$r = r_{\text{min}} + \frac{(r_{\text{max}} - r_{\text{min}}) \times (255 - I_{\text{origin}})}{255} \tag{5.1}$$

Stipple size r is linearly proportional to the original intensity I_{origin}; a minimum size stipple is placed at a site of intensity 255 (completely white), up to a maximum size stipple when the intensity drops to zero (completely black). Originally, the intensities are the same in uniform region and are gradually changed in areas for smooth surfaces. However, after our algorithm runs error diffusion for a few times, the intensities in the uniform regions will not be the same. The intensities in the areas for smooth surfaces will become steeply changed after diffusion, not showing smooth surfaces any more. If we use modified intensities to determine the stipple size, the

stipples in uniform regions will not look even and the stipples in smooth regions will not look gradually changed at all.

The stippling algorithm, as presented so far, still creates huge numbers of stipples. We propose to adjust the error carried forward in the diffusion process in such a way as to reduce stipple counts. Previously, we only draw stipples when the current intensity value of the pixel is below the threshold. Dark pixels, those with intensity less than 128, will initially be below the threshold. After error diffusion, dark pixels may be raised above the threshold because they may accumulate positive error from nearby pixels. Conservely, light pixels that will be initially above the threshold will be reduced below the threshold if sufficient negative error is accumulated, and a stipple will be placed. We propose a shrinking and exaggeration function to adjust the error. When the error is negative, the strategy is to reduce the magnitude of the assigned error (called shrinking), and when the error is positive, the strategy is to increase it (called exaggeration). Negative error occurs in light areas and brings darkening. Shrinking reduces negative error, which limits production of stipples in lighter portions of the image. Lightening happens when positive error occurs. Exaggeration increases positive error which exaggerates the impact of each stipple. The area near the stipple thus is lightened by a greater proportion than is actually warranted for faithful tone reproduction, which limits the production of stipples in darker portions of the image. Since we still use a priority-based scheme and error diffusion adheres to the policy of preferentially assigning more positive error to lighter pixels and more negative error to darker pixels, shrinking and exaggeration adjustment does not prevent the method from preserving structure in the stipple distributions. We employ a nonlinear gamma correction for the shrinking and exaggeration functions. The use of gamma correction can raise the input value to the G_- or G_+ power, providing a contrast-aware adjustment. Equation 5.2 calculates s_{xy} at pixel (x, y), a spatially-related adjustment factor for negative and positive errors that are independently controlled, using G_- and G_+ respectively. The error from pixel (x, y) is represented as e_{xy}.

$$s_{xy} = \begin{cases} (1/r)^{G_-} & \text{if } e_{xy} < 0 \\ (r)^{G_+} & \text{otherwise.} \end{cases} \tag{5.2}$$

To have a large reduction of the stipple number, we can increase the powers G_- and

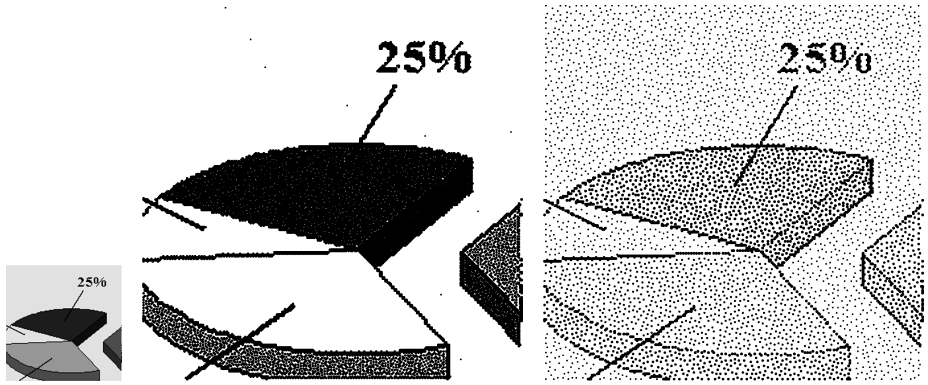
G_+, which bring less negative error for nearby pixels (producing fewer stipples) and more positive error (also producing fewer stipples). The adjustment for parameters G_- and G_+ (greater than 1.0) can control the degree of shrinking and exaggeration for light and dark regions respectively, which gives us flexibility in density control. The outcome of the reduction of stipples makes the resulting stippling lighter than the original image. However, the process retains structure and contrast.

The size of a pixel depends on image resolution. When a stipple is drawn to replace a pixel, we have to consider the size of the stipple. The error from placing a stipple is not simply the intensity from the pixel. A stipple with a large size takes more space. In Equation 5.3, the introduction of a correction term e_0 compensates for the case when a stipple is placed. The intensity updates become

$$I(m,n)' = I(m,n) + \hat{w}_{mn} \times (e_{xy} + e_0) \times s_{xy}, \tag{5.3}$$

where e_0 is given by $e_0 = (A_{\text{stipple}} - A_{\text{pixel}}) * k$. The updated intensity of the pixel at (m,n) after error diffusion is $I(m,n)'$. The values A_{stipple} and A_{pixel} represent the area of a stipple and of a pixel respectively. Notice that the adjustment of shrinking and exaggeration cannot reduce the number of stipples in black areas because the error is zero and the new function cannot affect the error. We introduce k, a user-adjustable parameter, to have an influence in black areas. In addition to the adjustment of black areas, extreme exaggeration of error could cause updated intensity values to fall outside the usual range of [0,255]. In that case, the intensities are clamped. Clamping will lose intensity. To reduce the lost intensities from clamping, we suggest to use a larger mask because the increments per pixel are smaller. Since exaggeration reduces stipple counts, the use of a larger mask size allows the full effect of exaggeration to take place and yields fewer stipples.

Figure 5.3 shows how the parameter choices affect the results visually. The notation $P(9,2,0,7)$ indicates the parameters $G_- = 9$, $G_+ = 2$, $k = 0$, and a 7×7 mask. Figure 5.3 (b) is obtained by using larger G_- and smaller G_+ and Figure 5.3 (c) employs smaller G_- and larger G_+. Both figures draw the image with reduced stipple counts, but the former lightens the lighter areas while the latter lightens the darker areas. This example shows how to adjust contrast in distribution. We suggest using $G_+ = G_-$ or G_+ a little smaller than G_-. Sparser distributions of stipples can be obtained by using a larger mask size. The value for k is usually very small.



(a) (b) $P(9, 2, 0, 7)$ (c) $P(2, 9, 0, 7)$

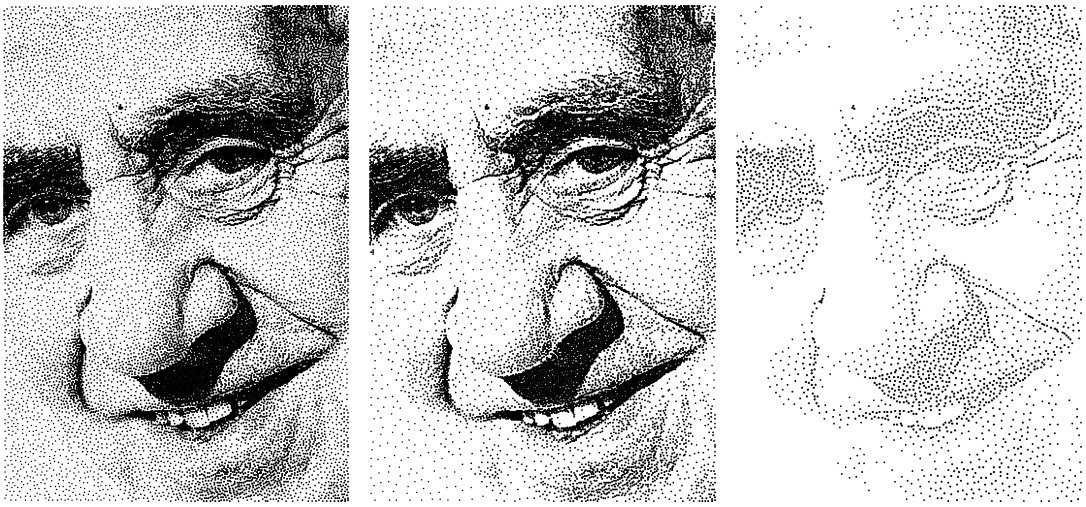
Figure 5.3: Parameter adjustment with reversing values of $.G_-$ and G_+ to show different contrast. (a) A larger original image is in Figure 3.12; (b) enhanced contrast with 12,075 stipples; (c) weakened contrast with 6,132 stipples.

Figure 5.4 shows another result of error adjustment on an image. This example demonstrates that the stipple count can be dramatically reduced without a major impact on structure quality. We use the same value of G_+ in both Figure 5.4 (c) and Figure 5.4 (d). There are 37,029 stipples in Figure 5.4 (c) with little shrinking or exaggeration of errors (small G_-). Figure 5.4 (d) uses fewer stipples with further shrinking or exaggeration of errors (larger G_-). In Figure 5.4 (e), we double the mask size, which reduces the number of stipples to 16,980. The reduction of stipples still is able to show the face wrinkles quite well. We show that a further reduction is possible using very large values of G_- and G_+ and with very large mask size in Figure 5.4 (f); at this point we still can discern the face even using only 3,085 dots. The use of k reduces the number of stipples for the black shadow under the nose. Procedure 2 shows details of the basic structure-preserving stippling method.

5.3.2 Tone Matching with Stipple Resizing

Reducing stipple counts makes the stippling appearance lighter because fewer stipples contain less black, all else equal. Making more dramatic adjustments to stipple size can improve the loss of tone in Figure 5.4. We can calculate stipple sizes in the way similar to the calculation by Secord [161]. The calculation is done after stipple locations have been established. Secord integrated input darkness over the Voronoi region of a stipple and set the size of each stipple accordingly.

(a) Original image (b) CAH (c) $P(5, 3, 0, 7)$



(d) $P(5, 5, 0, 7)$ (e) $P(5, 5, 0.1, 15)$ (f) $P(10, 10, 0.1, 15)$

Figure 5.4: Basic structure-preserving stippling method: transition from CAH to stippling with stipple budget decreasing. (a) Original image; (b) CAH with 95,156 black pixels and 87,116 white pixels; (c) 37,029 stipples; (d) 24,370 stipples; (e) 16,980 stipples; (f) 3,085 stipples.

Figure 5.5 displays the result of our resizing process. Figure 5.5 (a) shows the stipple regions and Figure 5.5 (b) gives the corresponding result with tone improved. Figure 5.5 (c) shows another resized stippling: however, large dots are used on the

Procedure 2 Basic Structure-Aware Stippling Method

```
1: Input: G_-, G_+, k, D, I_origin [I_origin, a given 8-bit image]
2: Output: stippleslist [A list of stipples with size information]
3: I ← I_origin
4: M ← false [false means unprocessed pixels; true means processed pixels.]
5: Heap ← BuildPriority(I_origin) [Sort pixels from the highest priority to the lowest
   priority]
6: while Heap is not empty do
7:    the pixel (x, y) with the highest priority p_hi ← Heap.pop()
8:    p(x, y) ← CalculatePriority(I(x, y)) [Equation 3.6]
9:    if p_hi ≠ p(x, y) [Current priority and old priority not equal] then
10:      Heap.update(p(x, y)) [Dynamically update priority]
11:   else
12:      if not M(x, y) [M(x,y) is a boolean value at position (x, y).] then
13:         r ← Calculatestipplesize(I_origin(x, y)) [Equation 5.1]
14:         if I(x, y) <= 128 [Intensity below threshold] then
15:            App ← 0 [Black]
16:            stippleslist.add(x, y, r) [Store the stipple position and the size]
17:         else
18:            App ← 255 [White]
19:         end if
20:         e_xy ← I(x, y) − App [Calculate error]
21:         Errordiffusion(x, y, e_xy, r, G_-, G_+, k, D, M) [Procedure 3]
22:         M(x, y) ← true [Processed this location]
23:      end if
24:   end if
25: end while
```

face, which is unattractive. Large dots evoke the connotation of blemishes and freckles. Although resizing stipples indeed improves the tone, there has been a detrimental impact on structure and visual appearance. This suggests that if very few stipples are used to depict an image content, large dots are not favored. Small stipples are better and hence tone loss is unavoidable under a very low budget when structure

Procedure 3 Errordiffusion($x, y, e_{xy}, r, G_-, G_+, k, D, M$) [$x, y$ the position, e_{xy} the error, r the stipple size, D the mask size, G_-, G_+, k, D parameters for density control, M the occupation map]

```
for all pixels I(m,n) under the circular mask with the center (x,y) and the radius D/2 do
    if not M(m,n) then
        r_mn ← the distance from (m,n) to (x,y)
        ŵ_mn ← CalculateWeight(I(m,n), r_mn, e_xy) [Equation 3.4]
        s_xy ← Shrinkorexaggerate(r, e_xy, G_-, G_+)[Equation 5.2]
        e_0 ← (A_stipple − A_pixel) * k [Correction]
        I(m,n) ← I(m,n) + ŵ_mn * (e_xy + e_0) * s_xy [Update]
        I(m,n) ← max(0, min(I(m,n), 255)) [Clamping]
    end if
end for
```

preservation is desired as well.

5.4 Variations on Masks, Priority Scheme, and Colors

We next explore variations including mask shapes, priority configuration, and colors to present diverse new styles for stippling such as irregular arrangements, stipple-based patterning, scratchboard, heightening, and painterly rendering. The same parameters for density control are available for all of those styles, which provides different levels of abstraction.

5.4.1 Variations on Masks

We propose a set of variations on masks: exclusion-based masks, line masks, and irregular masks. Exclusion-based masks, mentioned in Chapter 4, are combined with density control in order to provide sparse distribution for screening effects. Line masks provide texture effects such as texture on rough surface. Irregular masks are used to create irregular distributions.

(a) $P(5, 5, 0, 7)$ (b) $P(5, 5, 0, 7)$ (c) $P(5, 5, 0.1, 15)$

Figure 5.5: Our size modification strategy. (a) stipple influence areas; (b) result of (a); (c) another result. See also Figure 5.4 (d) and (e), examples without resizing stipples for comparison.

Exclusion-based Masks

As we mentioned in Chapter 4, the goal of designing exclusion-based masks is to create interesting patterns. If a 45-exclusion mask is used, the error will not propagate into the pixels along the diagonal direction; thus, these pixels will retain their original intensities and priorities. In general, and especially in uniform regions, error diffusion reduces the priorities of affected pixels. Thus, the excluded diagonal pixels will tend to be processed earlier, with all error being pushed to the off-diagonal pixels; visible diagonal structures then appear in the output, as we can see in Figure 5.6 shows diagonal patterns. The excluded pixels in light regions are more likely not to be chosen as stipples. Sparse distributions can still display structure and patterns in the output image, shown in Figure 5.6 (right). We can create patterns based on the angles and the number of exclusion directions; see Figure 5.7. Even low stipple counts can indicate both patterns and content clearly.

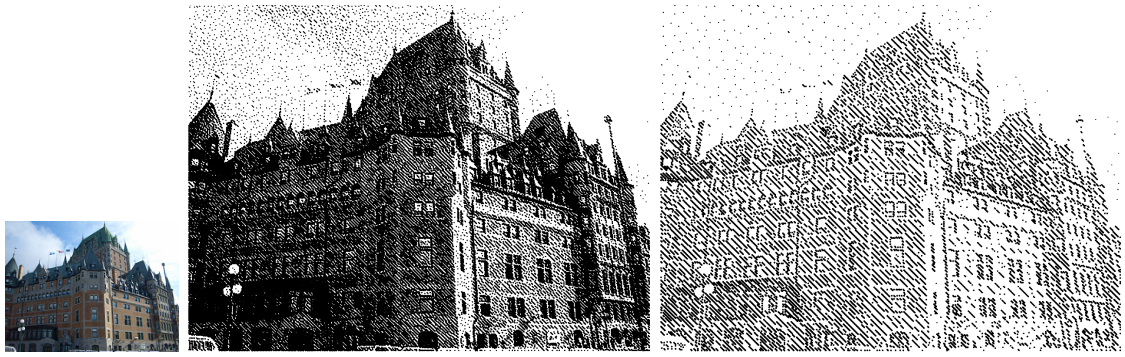
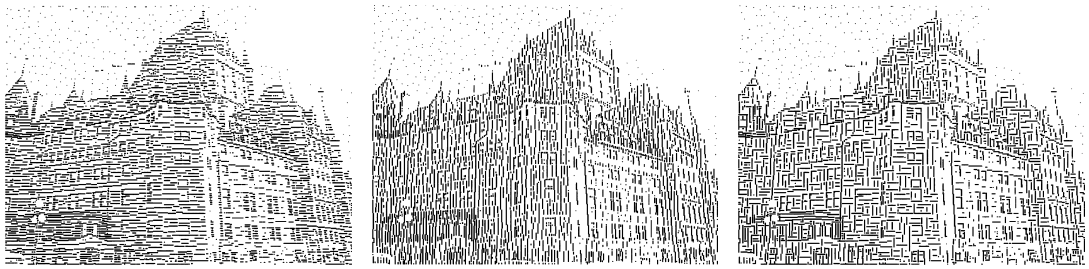


Figure 5.6: Two different levels of abstraction for 45°-exclusion. Left: Original image; middle: dense stippling with $P(5, 3, 0, 7)$; right: sparse stippling with $P(6, 6, 0, 21)$.



(a) (b) (c)

Figure 5.7: Very low numbers of stipples. Three examples for exclusion-based masks: (a) Exclude one direction with 0°; (b) exclude one direction with 90°; (c) exclude two directions with 90° and 0°.

Line Masks

Previously, an exclusion-based mask excludes some pixels of a circular mask. Directional exclusions can generate interesting patterns. A line mask excludes all pixels except those along the line. An example of a line mask is shown in Figure 5.8.

When a pixel is to be processed, we preserve the pixels along the edge direction θ in the mask. The pixels not aligning along the edge direction will not be involved in the process of error diffusion. Figure 5.8 shows a line mask with a user-defined length L. The system determines the output for this pixel: black or white. Error is diffused to the pixels of the line mask. Figure 5.9 (c) and (d) display the results from line masks. Both present the bird with interesting texture, like the effect of drawing on a rough or bumpy surface. In Chapter 4, exclusion-based masks can promote patterns

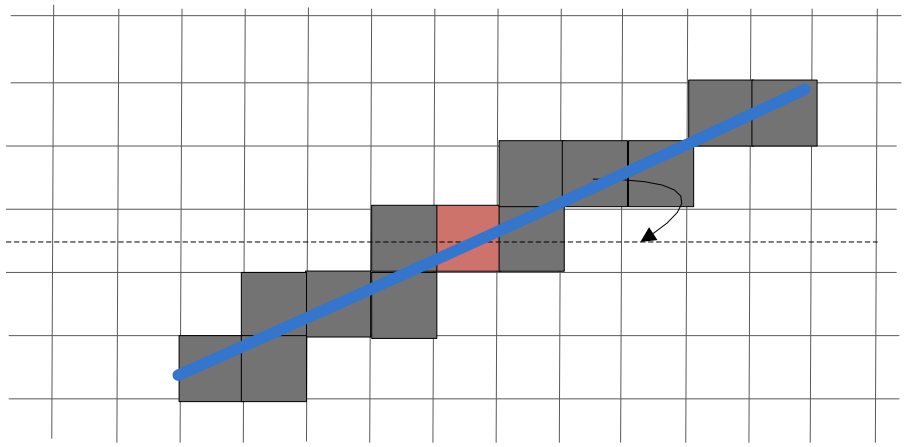
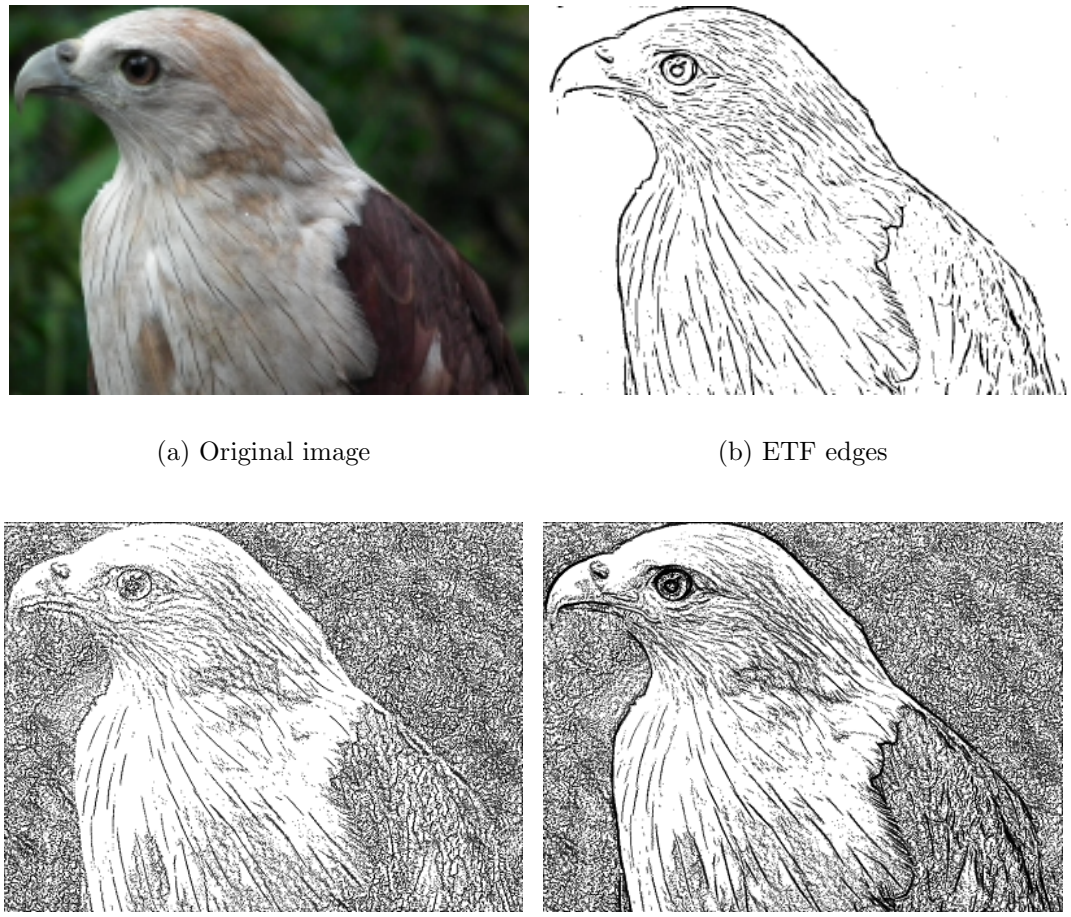


Figure 5.8: Representation for a line mask.

that are defined by excluding pixels in certain directions. The pixels excluded in a mask are likely to be black in dark regions. All pixels in the line mask are excluded, including the pixels close to the current pixel. Using a line mask, our algorithm may choose a pixel near to the current pixel as the next candidate, which has been avoided in previous exclusion variations; however, it presents opportunities to create texture effects. Figure 5.9 (c) shows the bird in an acceptable way and the structure quality could be enhanced by multiple stages as well, shown in Figure 5.9 (d). This idea of line mask is an experiment on pixel management to create textures.

Irregular Masks

Stippling distributions created by artists are random and irregular. The Eden growth model [29] can generate irregular regions by using random accumulation. We propose an Eden-like growth scheme for irregular mask generation to break the excessive regularity of stipple distribution. Instead of a circular mask, a fixed-size mask is generated by randomly adding 8-connected pixels until the target pixel count is reached. Figure 5.10 demonstrates how a 26-pixel irregular mask evolves with the Eden growth. The blue pixels in the figure are future pixels, which may be chosen for the next selection. The scheme randomly clusters pixels on the boundary. The outcome of



(a) Original image

(b) ETF edges

(c) Without ETF promotion

(d) ETF edges promoted

Figure 5.9: Bottom row shows texture from line masks.

the growth is a random and irregular shape. Each pixel employs a different random shape of a mask to diffuse error. We show the difference between circular masks and irregular masks in Figure 5.11. Notice that the face of the clock on the right side has a less ordered stipple distribution compared to the image on the left side which uses circular masks. In regions of uniform color, the irregularity in spatial distribution of error produces an uneven distribution of stipples. Importantly, the irregularity does not reduce the ability to preserve structure because irregular masks still distribute errors in a contrast-aware way. It takes around 50 seconds for our system to process the clock. Though the generation of an irregular mask for each pixel is time-consuming, the idea of varying shapes for masks might inspire another approach for hand-drawn stippling and also might give an initial clue to introduce stipple shapes into an error

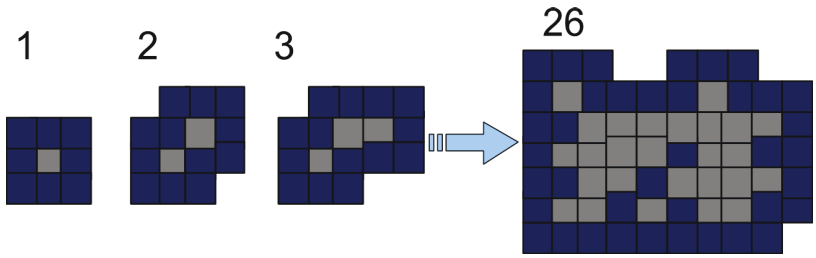


Figure 5.10: An example showing the growth of a 26-pixel irregular mask for a pixel.

diffusion scheme. This exploration also shows the flexibility of our system.

5.4.2 Variations on Priority Scheme

In Chapter 4, we described a multiple-stage process for the priority scheme for screening. Stippling also can be processed in multiple stages. In each stage, we can apply density control independently to give varied levels of emphasis. Unimportant regions of the image can be processed with more aggressive stipple reduction, while a larger number of stipples can be dedicated to important areas. The system has parameters $G_{1,-}$ and $G_{1,+}$ to control dot density in the first phase; in general, we write $P_j(G_{j,-}, G_{j,+}, k_j, D_j)$ for the jth stage. Using separate parameters for different stages provides flexible density control over each stage. Users can promote interesting regions with low parameter values and deemphasize uninteresting stages with higher values. In the following, we present three specific ways of exploiting multiple priority levels.

Promotion of ETF Edges This is a two-stage process to promote ETF edges, called ETF-enhanced stippling. In the first stage, the priority set is drawn from pixels on the edges, which are obtained by using the flow-based difference of Gaussians on the edge tangent field (ETF) [71]. We show two two-stage results in Figure 5.12 to demonstrate two different stipple distributions with ETF edges promoted. Both results express the ETF edges clearly.

Promotion of Geometric Curves: Figure 5.13 shows individual results of each stage in a three-stage process. We show each priority set in Figure 5.13 (a). The first priority set S_1, in red, is a geometric arrangement of lines; the second set S_2, in blue,

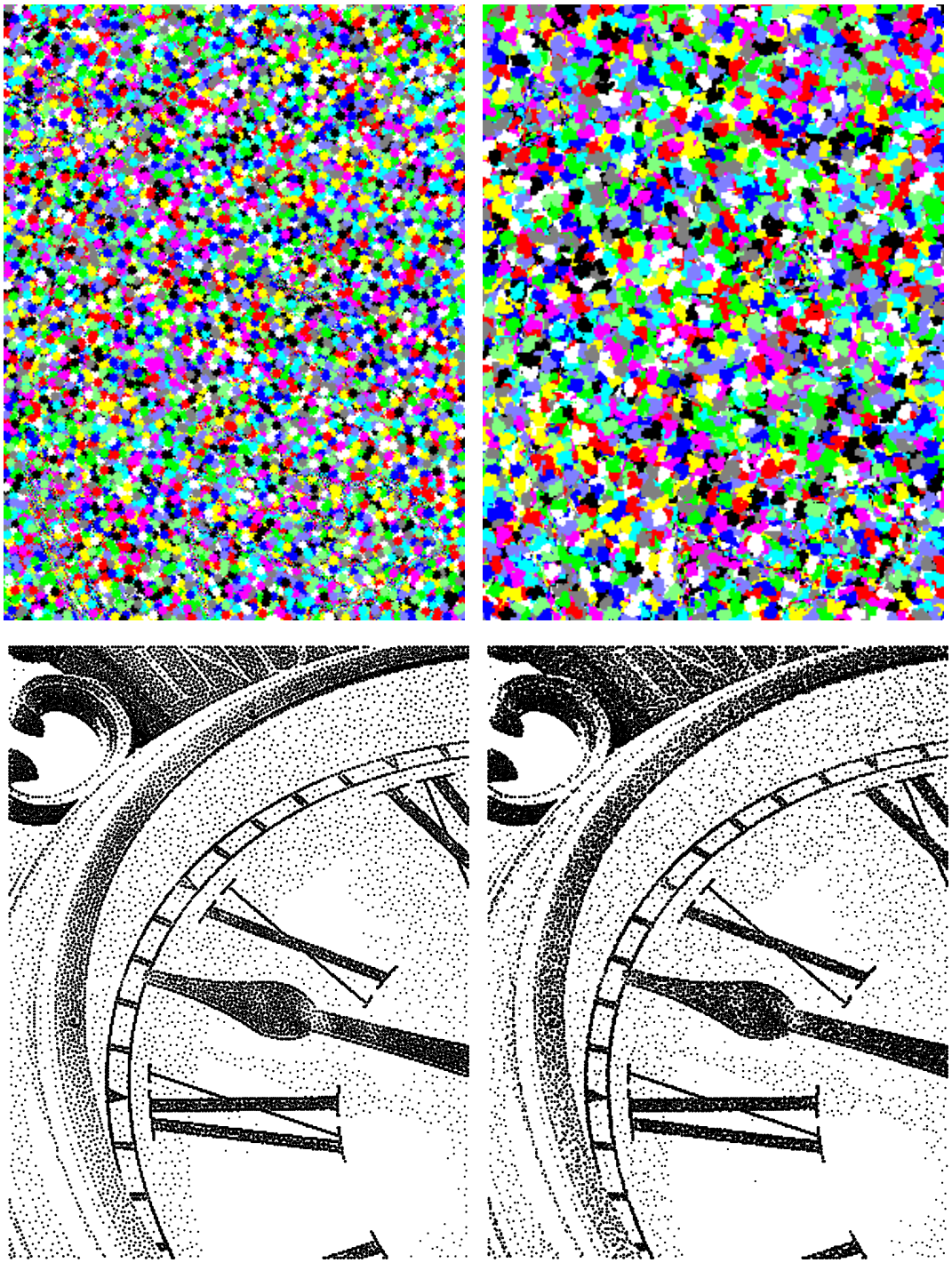
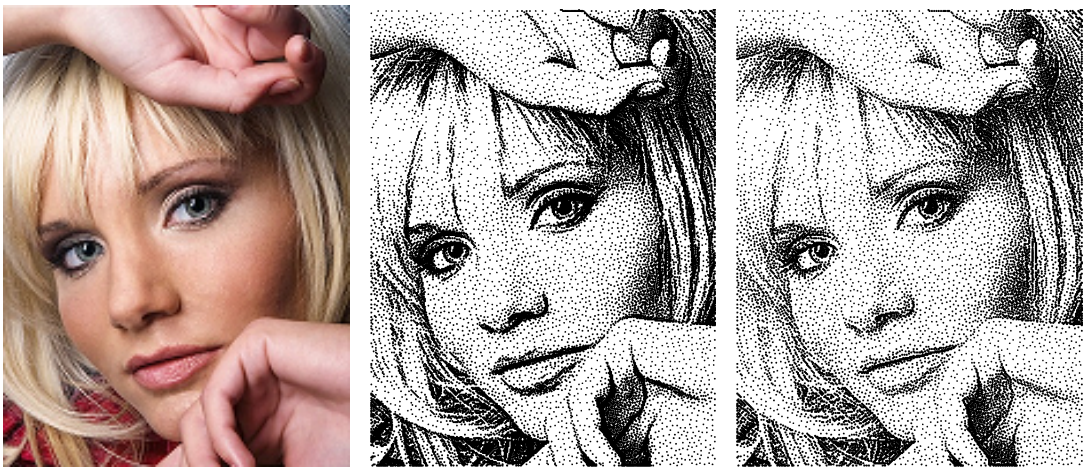


Figure 5.11: Comparing stippling from both circular and irregular irregular masks; both use $P(5, 3, 0, 7)$. Left: the stippling result from circular masks. Right: the stippling result from irregular masks.



(a) Original image (b) Dense ETF edges (c) Sparse ETF edges

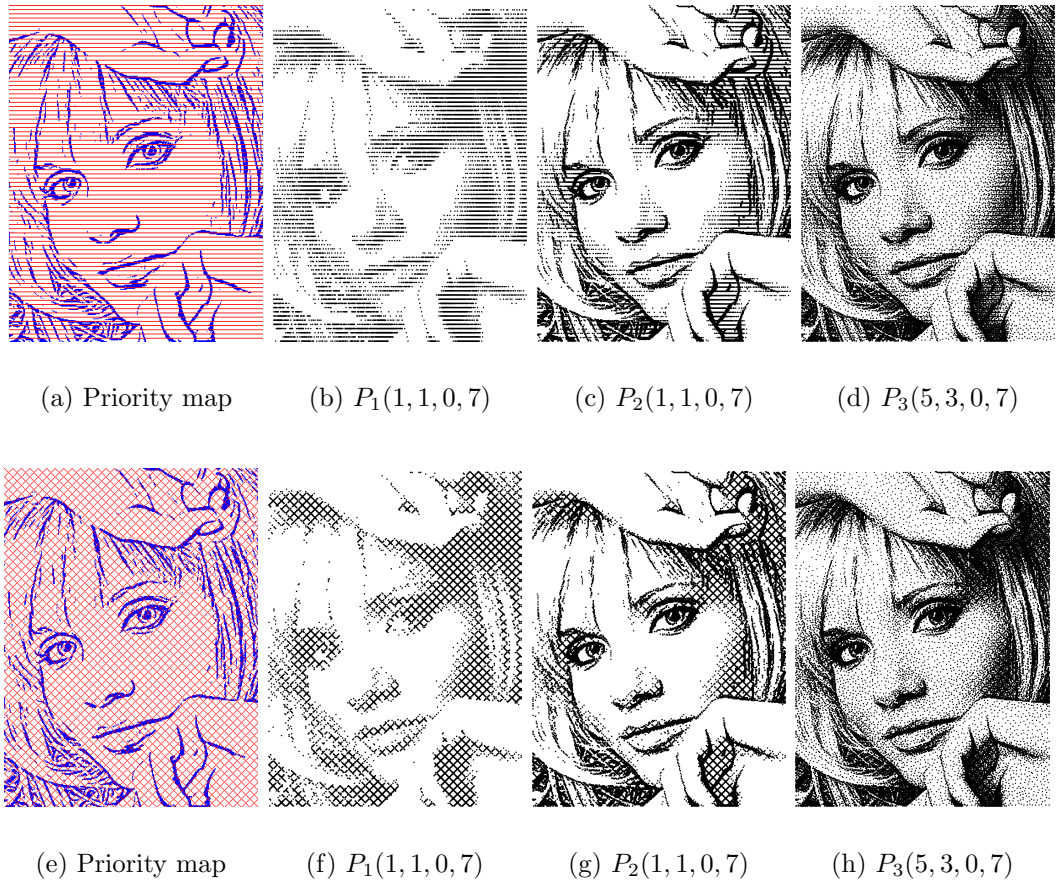
Figure 5.12: ETF-enhanced stippling for two densities after a two-stage process. (b) $P_1(1, 1, 0, 7)$ and $P_2(5, 3, 0, 7)$; (c) $P_1(3, 3, 0, 7)$ and $P_2(5, 3, 0, 7)$.

is the collection of ETF edges; and the third set S_3, in white, is the remaining pixels. Figure 5.13 (b) shows an intermediate result after processing the first set. Although the number of stipples in the output after the first stage is very low, the tone and structure of the woman is discernible. Figure 5.13 (c) shows the second-stage result. Figure 5.13 (d) shows the final result that displays both horizontal lines and ETF edges. The bottom of Figure 5.13 shows the procedure for another example with a priority map using crossed lines.

Promotion of Textural Edges: To introduce a variety of patterns from texture images into stippling results, one approach is to take the priority map from an input image. Different examples with different textures are shown in Figure 5.14 by using three stages to process textural edges, ETF edges, and the remaining pixels in order.

5.4.3 Variations on Colors

The structure-preserving scheme for stippling provides good quality in abstracting content from an image. We here adapt our stippling algorithm to different artistic styles: *heightening*, *pointillism*, and *scratchboard*, by using color variations.



(a) Priority map (b) $P_1(1, 1, 0, 7)$ (c) $P_2(1, 1, 0, 7)$ (d) $P_3(5, 3, 0, 7)$

(e) Priority map (f) $P_1(1, 1, 0, 7)$ (g) $P_2(1, 1, 0, 7)$ (h) $P_3(5, 3, 0, 7)$

Figure 5.13: Screening with stipples from geometrical curves by a three-stage process. (a) A map for priority configuration; (b) the first-stage result; (c) the second-stage result; (d) the final result.

Heightening

Heightening is a style with a striking high-contrast effect: dark shades occupy most of the image on a neutral background, and light shades are used sparsely. We propose to create a heightening effect by using the basic stippling method twice. First, we obtain black dots based on the original image. The first process builds the majority of stipples in black, providing tonal and structural components. Next, the stippling process is done on an inverted image of the original image without using shrinking and exaggeration. The second stippling process creates the stipples for the highlights. Stipples are created in priority order; we take the 4The white stipples provide highlights for bright areas. We set the background color with gray level 128, an intermediate color value. Figure 5.15 demonstrates the heightened effects, which give

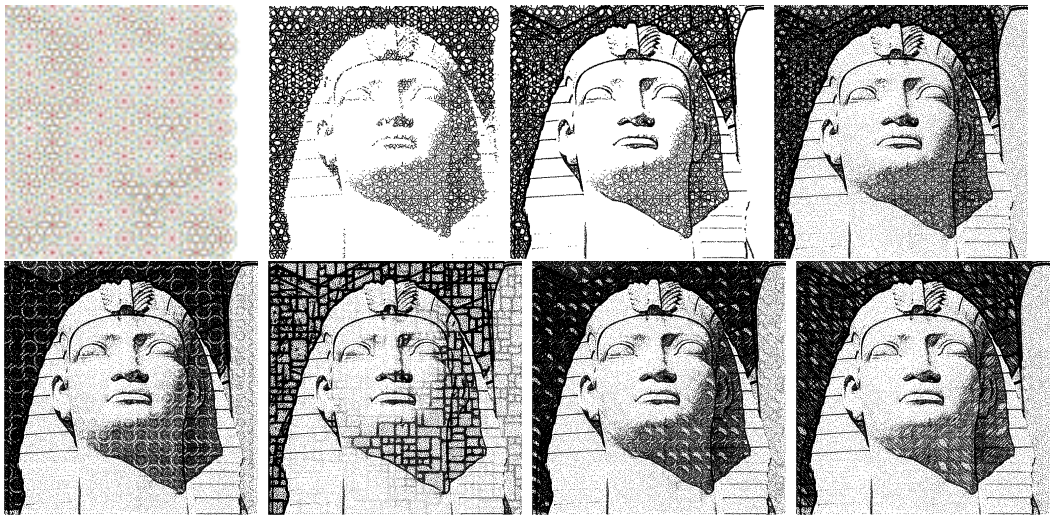


Figure 5.14: Screenings with stipples from texture input by a three-stage process.

some details a shinier appearance. To our knowledge, our work is the first to treat heightening in the context of computer-generated stippling.

Figure 5.15: Heightening effects from color variations of basic stippling.

Pointillism

We color the dots instead of only using black or white, yielding a pointillism effect. A result is shown in Figure 5.16, which is created by first computing a stippled image and then coloring each stipple with the original image color. The overlapping dots

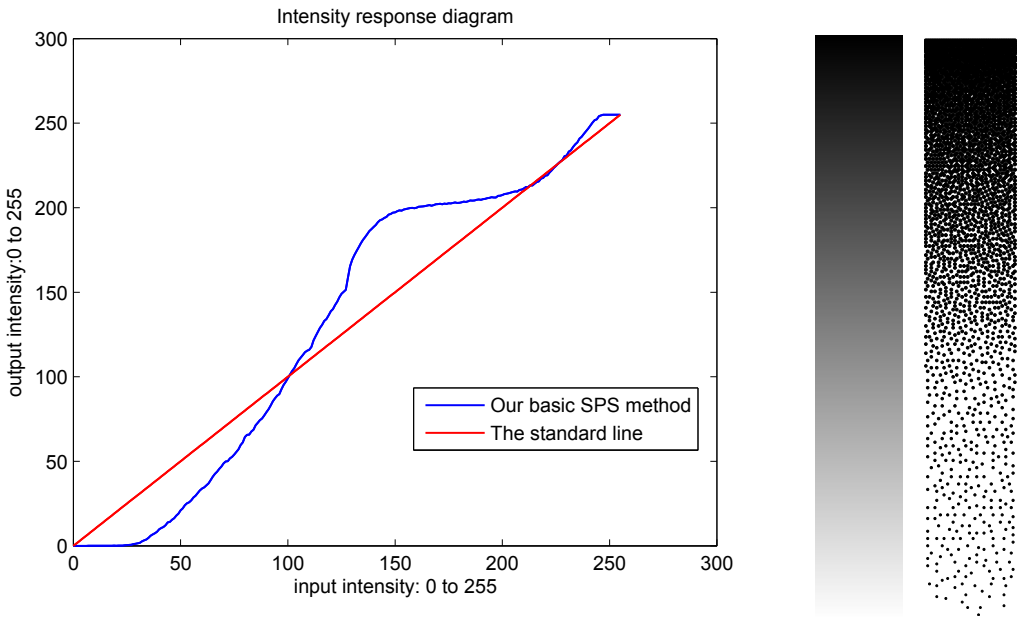
Figure 5.16: Pointillism effects by colored dots of a basic stippling.

with different colors generate the effect. The effects by color variations are similar to styles created by Jang and Hong [65] and Yang and Yang [196].

Figure 5.17: A scratchboard effect with dots from edge-exclusion masks.

Scratchboard

Scratchboard is a technique used by artists to generate highly detailed and textured artwork, made by scratching a thin black layer to uncover white clay beneath. Our edge-exclusion method is very good at conveying structural content. Highly detailed structures and textures are common in the artistic medium of scratchboard. We proceed by negating the input image and then use white stipples on a black background to represent it. In Figure 5.17, the highly-detailed hair textures for the old man are clearly preserved.



(a) (b) (c)

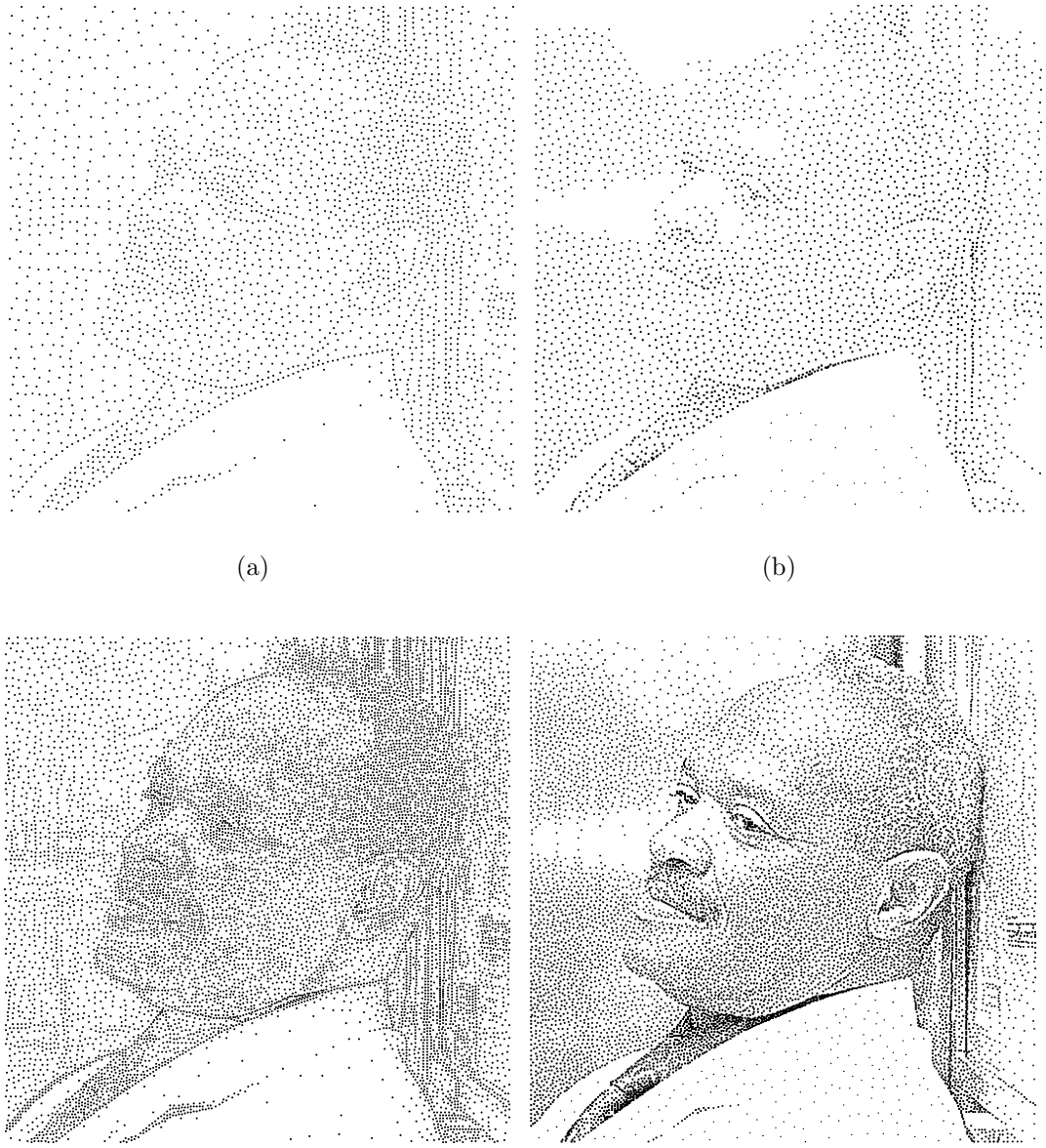
Figure 5.18: Tone analysis: (a) the red line shows exact tone matching and the blue line shows our intensity response; (b) original ramp; (c) our stippled ramp by SPS.

5.5 Results and Discussion

We compare and analyze our basic stippling method as well as its variations in this section. The goal of this section is to help evaluate our stippling approach. Our stippling results are generated by using our basic stippling method with parameter $P(5, 3, 0, 7)$ unless otherwise stated.

We show the analysis of tone quality in Figure 5.18. The intensity response diagram is generated by producing a stippled result for each greylevel and calculating the percentage of blackness for each stippling image. The intensity response curve shows, for every input tone level, the resulting tone level after SPS. The curve shows continuous contrast enhancement: light input tones have even lighter output, while darker ones are darkened even further. This is a necessary property given our structure-based objectives. Visually, the effect is quite good in Figure 5.18 (c).

Structure preservation is our main concern, not tone matching. The tower images in Figure 5.1 show that our stippling method preserves structure better than previous stippling methods. Figure 5.19 compares against the results from Mould [123], who



(a) (b)

(c) (d)

Figure 5.19: Comparison with Mould's method. Top left: Mould's method, approximate 3,000 stipples; Top right: our stippling method, 3,113 stipples, $P(10, 10, 0, 17)$; lower left: Mould's method, approximate 11,000 stipples; lower right: our stippling method, 10,550 stipples, $P(6, 6, 0, 17)$.

shares our goal of structural preservation in stippling. The comparisons indicate that our stippling method is able to achieve significantly better qualities of tone reproduction and structure preservation than Mould's method, both with high and

very low budget of stipples, for an overall much better effect.

We apply the mean structural similarity index measure (MSSIM) [181] to quantify the structural difference between the stippling result and the original grayscale image. Table 5.1 gives the measurement scores for structure similarity; Table 5.2 shows tone similarity by peak signal-to-noise ratio (PSNR). For both MSSIM and PSNR, higher values indicate higher similarity. To make a fair comparison, all test images are from Kim et al.'s paper [84] and all results are first produced as vector graphics and then converted into raster images (using Inkscape) at the original image resolution. Since Secord's method can control the exact number of stipples used, Secord's results have the same number of stipples as ours. Thanks to the structural preservation, both our basic stippling method with $P(5, 3, 0, 7)$ and ETF stippling method with $P_1(0, 0, 0, 7)$ and $P_2(5, 3, 0, 7)$ have higher MSSIM values than other previous stippling methods, thus higher structure similarity. Apart from Secord's method which was carefully honed to match tone exactly, we have better tone matching than others. We also found an interesting comparison between SPS results and contrast-aware halftoning (CAH) results. Although the number of pixels in CAH halftoning is much larger than the number of stipples in SPS, Table 5.1 shows that higher structure similarity is achieved by our stippling method.

Image	baby	balloon	clock	kid	man	w1	w2	w3	w4
Our basic stippling	0.29	0.52	0.52	0.58	0.42	0.33	0.59	0.55	0.15
ETF-enhanced	0.30	**0.55**	**0.55**	**0.61**	**0.44**	**0.36**	**0.63**	**0.56**	**0.15**
Stipple resizing	0.05	0.44	0.30	0.20	0.10	0.12	0.30	0.17	0.08
Kim et al.'s	**0.33**	**0.55**	0.46	0.33	0.38	0.17	0.49	0.32	0.18
Secord's	0.13	0.52	0.32	0.20	0.12	0.10	0.34	0.29	0.07
Martín et al.'s	0.07	0.45	0.19	0.07	0.06	0.07	0.18	0.07	0.08
CAH	0.04	0.45	0.28	0.14	0.06	0.12	0.26	0.13	0.09

Table 5.1: Structural similarity measured by MSSIM; the maximum values in each column are shown in bold. w is short for woman.

The proposed SPS method is an evolved error diffusion scheme. The computational cost depends on the image size and the mask size, not image content. Error diffusion processes each pixel once, which ensures an upper bound on computational complexity. Our SPS method is based on error diffusion hence has high efficiency and fast speed. This stippling scheme adapts the contrast-aware halftoning method

Image	baby	balloon	clock	kid	man	w1	w2	w3	w4
Our basic stippling	12.4	17.7	14.0	13.7	13.7	13.0	15.1	16.8	14.3
ETF-enhanced	13.1	19.0	15.4	15.2	14.6	14.4	16.5	17.4	14.5
Stipple resizing	21.4	19.0	21.3	22.5	24.2	21.4	23.4	20.8	18.3
Kim et al.'s	13.0	10.8	13.3	16.3	12.8	10.9	14.3	13.8	8.7
Secord's	**27.4**	21.3	23.8	**29.5**	28.4	23.8	25.7	21.4	18.7
Martín et al.'s	8.0	12.9	9.2	8.6	8.2	8.2	9.9	9.9	9.7
CAH	23.9	**24.5**	**28.4**	26.1	**32.4**	**31.7**	**31.8**	**30.9**	**28.6**

Table 5.2: Tone similarity measured by PSNR; the maximum values in each column are shown in bold.

Image	image size	stipple count	time (s)
Figure 5.1 (d)	395 by 456	30,209	6.2
Figure 5.3 (b)	287 by 286	12,705	3.1
Figure 5.3 (c)	287 by 286	6,133	2.9
Figure 5.4 (c)	356 by 512	37,029	6.0
Figure 5.4 (f)	356 by 512	3,085	5.6
Figure 5.19 (b)	512 by 512	3,271	9.4
Figure 5.19 (d)	512 by 512	10,550	11.9

Table 5.3: Timing results for some of the stippling processes.

by introducing a shrinking and exaggeration function to adjust the amount of error diffused. Our stippling method runs in the same priority-based scheme as the CAH method. The difference is in the extra calculation of the shrinking and exaggeration function. Given an image with N pixels, the CAH method runs $NlogN$ steps. Our stippling method runs $NlogN + cNLogN$ steps, where $NlogN$ is the same steps as the CAH method and $cNLogN$ are the additional steps for calculating the new non-linear function. The symbol c represents constant steps used in the new calculation for each pixel. Thus, SPS shares the same complexity with CAH, $O(NlogN)$. Most of our results are produced in only a few seconds to tens of seconds based on an Intel(R) Core(TM) i3-2350M CPU @2.3GHz with 8GB RAM. See Table 5.3 for the timing details. The use of irregular masks took longer time than other variants; this variant takes nearly a minute to obtain an irregular stipple distribution. The fast speed as well as good structural quality is a distinct advantage over previous stippling methods, which usually cannot achieve both. We cannot provide the details

of time comparison with other stippling methods because we do not have the exact implementations of previous stippling methods and their timing reports are based on specific machines, which we cannot obtain. We only give a rough report to provide a basic impression of time comparison. Secord [161] requires 20 minutes to draw up to 40,000 stipples. The timing from Secord's method is based on much older hardware, which we cannot obtain for comparison. Kim et al. [84] reported that placing 8,000 to 12,000 stipples takes several minutes. Mould's and Martín et al.'s methods [117, 123] runs in seconds and are a little faster than our stippling method, but cannot achieve the same quality of structure and tone similarity in stippling. One limitation of our stippling method is that density control through adjusting nonlinear parameters is indirect. We cannot have control over the exact number of stipples. In summary, our new stippling method provides very good structure and also suggests the original image tone, while attaining high processing speed.

Notice that most of the effects such as screening with dots, heightening, and scratchboard are new for stippling. Thanks to the structure awareness, the priority-based scheme, and flexibility in style transition, stippled results with diverse effects still present the content details nicely. The styles we covered in this chapter look better than other automatic stippling methods in terms of structural preservation. The variations includes control over masks, the priority-based scheme, and colors. We introduce exclusion-based masks, irregular masks, and line masks for mask variations and design a multiple-stage process for the priority-based scheme. We show various artistic effects with good structure preservation. The multi-stage process can use texture images for promoting patterns in stippling, which mean our stippling can show numerous different patterns. Color variation provides varied effects for heightening, scratchboard, and pointillism.

5.6 Summary

In this chapter, we extended Prioritized CED to structure-preserving stippling by non-linear shrinking and exaggeration function to control the number of dots. This nonlinear error adjustment function can reduce stipple counts to a very low number but still present structure well. Thanks to the high quality of structure preservation and flexible priority-based scheme, our diverse evolved styles presented abstracted

content clearly.

This work applied for stippling has been published as follows:

- Hua Li and David Mould. Structure-preserving stippling by priority-based error diffusion. In *Proceedings of Graphics Interface 2011*, Graphics Interface (GI '11), pages 127–134. Canadian Human-Computer Communications Society, 2011

So far, based on Prioritized CED, halftoning, screening, and stippling work well for enhancing important features. The next chapter is going to demonstrate how this framework can be adapted for generating line art.

Chapter 6

Line Art

6.1 Introduction

The previous three chapters introduced and demonstrated Prioritized CED. This chapter explores Prioritized CED for lines. The task becomes to make use of line length and direction to help convey important information.

The major interest for this whole thesis is to preserve structural details in abstracted images. A single dot cannot represent orientation. We have to align a set of dots along a direction to display the orientation. We are interested in creating simplified line drawings, which have long and few lines, providing a clean appearance. The long lines depict object silhouettes, creases, and textures. The necessary tone is emphasized with some small lines. The task for the simplified line drawing is to determine the major structure for an image and to add extra lines for shading. However, the large-scale structure is not easy to obtain.

We use SPS to create drawings with short or long lines. Drawings with short lines are directly created by replacing each stipple with a short line, in which the direction of the line is obtained from the edge tangent field. To generate drawings with long lines, we introduce pixel clustering and employ skeletonization. Pixel clustering can cluster sets of pixels to form lines and skeletonization provides a simplified representation of the major structure of the image content. In Prioritized CED, we process an image pixel by pixel based on priority order. For line art, we propose pixel clustering that processes a set of pixels all at once; this set of pixels will align along the edge direction to form a line. The skeletonization technique finds Voronoi-based medial axis from ETF edges to obtain a clean simplified structure of an image. We demonstrate three applications using pixel clustering and skeletonization: simplified line drawing, emphasized line drawing, and a combination of stipple and lines. Simplified line drawings are drawings with very few long lines to display image content. Long lines are first drawn on the necessary content. We also add few short lines to

emphasize the important shading. Our automatic method for creating simplified line drawing clusters a set of pixels to form lines and finds the medial axis of objects for a simplified representation, which extends structure-preserving stippling. We first show a transition from stippling to line art and then propose a method for simplified line drawings. Emphasized line drawings, which thicken the edges, use more lines than simplified drawings. The combination of stippling and line art is a style using lines to display image structure and stipples to show shading. Both emphasized drawings and the combination of stippling and lines are created by applying error diffusion to a trail of pixels and producing a thin version of the image content. The drawings we create inherited good structure quality from structure-preserving stippling.

Contributions. In image space, it is difficult for automatic methods to generate line drawing with long lines and with good structure preservation. Our contribution is in proposing the techniques of pixel clustering and skeletonization, which can preserve the large-structure of an image with a few long lines.

6.2 Drawing with Short Lines

Stippling can be converted into line drawing by directly changing each dot to a line. There are two kinds of stipples: some stipples represent the key structures such as silhouettes, discontinuous edges, and textural edges; others are used to describe tone or shading. For stipples on edges, we draw a straight line following the edge tangent field, which roughly outlines the objects and important features in the image. The remaining stipples can be drawn in one of two ways. One way is to draw lines along the field direction; the other way is to draw as cross-hatching. The cross-hatching uses one direction following the field and an extra direction perpendicular to the field.

Figure 6.1 presents drawings with short lines applied to a screenshot of a 3D model. The line drawings for the bunny image use the direction of the field to align lines. Both our drawings in the middle and on the right of the figure with many short lines nicely show the object silhouettes and display shading using directional lines. Figure 6.2 shows another example similar to hatching with colored lines. The hatching in Figure 6.2 (a) forms the cheek, eyes, nose, and lips accurately, while underlying patterns with parallel lines nicely construct the face surface; a small disturbance on

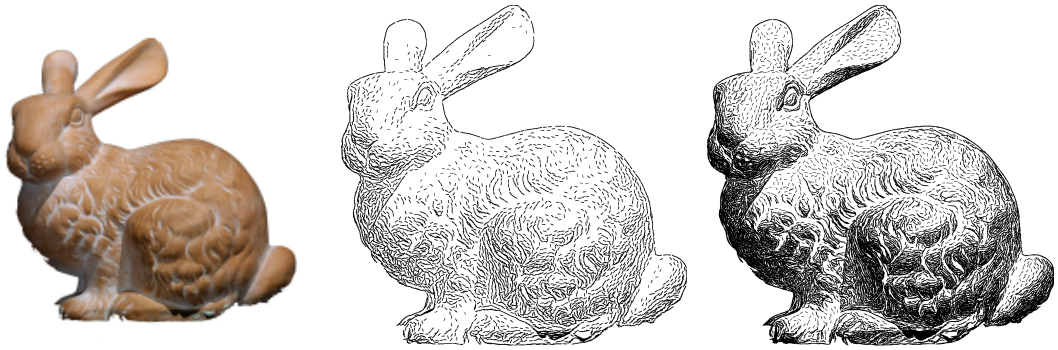


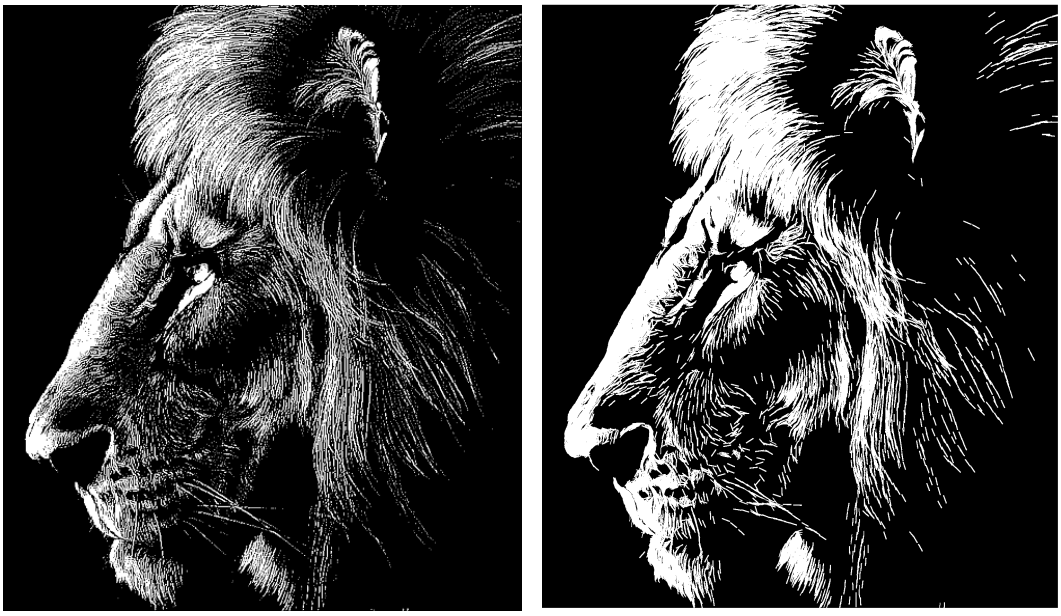
Figure 6.1: Line drawing with many short lines.

(a) (b)

Figure 6.2: Variations on line drawings with many short lines.

the extra direction in Figure 6.2 (b) provides cross-hatching effects similar to hand-drawn effects that have imperfect and irregular appearance.

Line-based Scratchboard: Although dot-based scratchboard in stippling application, shown in Chapter 5, is very impressive, real scratchboard pictures are expressed with beautiful flow of lines or curves. To draw line-based scratchboard, we first invert the original image. After generating stippling for the inverted image, the algorithm draws a straight white line replacing each dot on a black background. Line directions follow the edge tangent field. Figure 6.3 (b) shows a line-based lion, showing the impression of hair flow more gracefully than the dot-based lion shown in Figure 6.3 (a). Another scratchboard example appears in Figure 6.4. The seal hair in Figure 6.4 is nicely preserved by greyscale lines, in which the color of a line is the pixel color.



(a) Stipples (b) Lines

Figure 6.3: Comparison of two scratchboard effects for a lion. (a) dot-based; (b) line-based.

6.3 Drawing Extended Lines

The previous section extends stippling to line drawing by first creating a stipple distribution and then replacing stipples with short lines. However, the resulting images contain a high number of short lines. In order to obtain long lines and reduce the number of lines used, we cluster a set of pixels aligning along the edge direction to form a long element. Error diffusion is then applied to the set of pixels after clustering. We use skeletonization on an edge image. The skeletons can indicate the most important feature of an image. The long lines from pixel clustering provide shading with lines. Both skeletons and the clustered lines give a simplified appearance for line drawings.

6.3.1 Techniques of Pixel Clustering and Skeletonization

The basic idea of pixel clustering is to cluster a set of pixels in a coherent way in order to produce lines or curves. We still use the framework of Prioritized CED. Contrast-aware error diffusion takes care of tone indication while preserving structure.

Figure 6.4: Scratchboard with lines in greyscale.

Instead of processing one pixel in each step, our method for line drawings processes a set of pixels along the edge direction centered around the pixel. This set of pixels may be extended further by tracing along the field, which forms a long line. The skeletonization method concentrates on creating very long lines by finding the medial axis from the ETF edges. Short lines could distract viewers and hence we remove isolated short lines. The long lines keep the content structure neat, which makes image contents clearer to viewers than the short lines. We next describe the technique of pixel clustering and skeletonization for generating simplified line drawing, emphasized line drawing, and combined art of stipples and line.

Pixel Clustering

I: Directional Coherence from Edge Tangent Field: The proposed pixel clustering still processes the entire image in priority order in the same way as SPS. However,

instead of processing a single pixel at each step, we decide the final color for multiple pixels simultaneously. For each pixel, we grow a trail tracing along the edge tangent field in two directions: one following the direction and another following the opposite direction. Each direction proceeds for a distance l. The set of the pixels in the trail are S. The output color is based on the the average intensity $\hat{E}_b$ of the pixels in S: $\hat{E}_b$ using Equation 6.1,

$$\hat{E}_b = \frac{\sum_{(m,n)\in S}^{|} S|I(m,n)}{|S|} \tag{6.1}$$

Note that we used the notation $|S|$ to indicate the number of pixels in S. If $\hat{E}_b > 128$, white is chosen; otherwise, black. In practice we use $l = 10$; a short line avoids excessive deviation from the original direction of the edge tangent field. At the same time, using a straight line allows sharp corners to be portrayed, which have been smoothed away in the edge tangent field itself.

II: Trail Extension: After we have a set of pixels for error diffusion, we can do trail extension to grow a long line. In line drawings, it is common to see a long line used to represent edges. To obtain a longer line or curve, the process extends the trail by continuing new line segments from black paths. White paths are not extended. The extended lines use the direction found from the edge tangent field in the original image. As shown in Figure 6.5, each endpoint of the spawned set attempts to extend a new line on both sides along the direction from the edge tangent field. In the figure, A_0 has two new candidates A_1' and A_1'', forming corresponding angles: α and β. The endpoint with the larger angle will be chosen to further extend the trail. In Figure 6.5, A_1' is chosen as the new endpoint A_1. If α is equal to β both with 90 degrees, we can pick either A_1' or A_1''.

In the same way, the other endpoint B_0 could extend a new endpoint B_1. When finishing the new extension, the system uses the average intensity of the pixels from the new extension and old extensions to determine the path color. If the average intensity is greater than 128, white is chosen for the path color; otherwise, black. When the new extension is a black path, it will be added to the previous path; otherwise it will not be added and the extension in that direction stops. The process terminates when there is no path which needs further extension or when the extension intersects with previous paths or with the image boundary.

After extending the path, the system calculates the error individually for each pixel

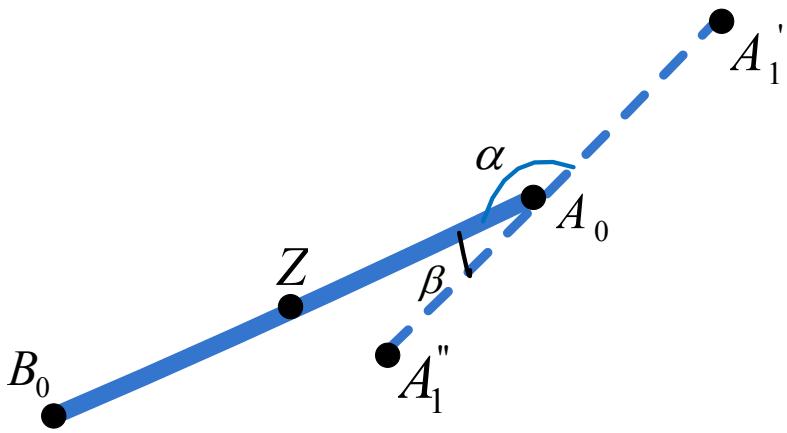


Figure 6.5: This figure shows trail extension in pixel clustering starting with a pixel Z, in which $A_1{}'$ and $A_1{}''$ are two candidates to extend A_0; and $A_1{}'$ is chosen because α is greater than β.

on the extended path and distributes the error. For each pixel on the extended path, error is diffused to the neighboring pixels under a circular mask using Procedure 3 in Chapter 5. The pixels on the path will not be adjusted during error diffusion. The extended paths form a long line. The SPS method can adjust the density of stipples and the preservation of structure for the image content using the parameters in the nonlinear function. Since we build our line drawing method on SPS, those parameters are available to our drawing method as well.

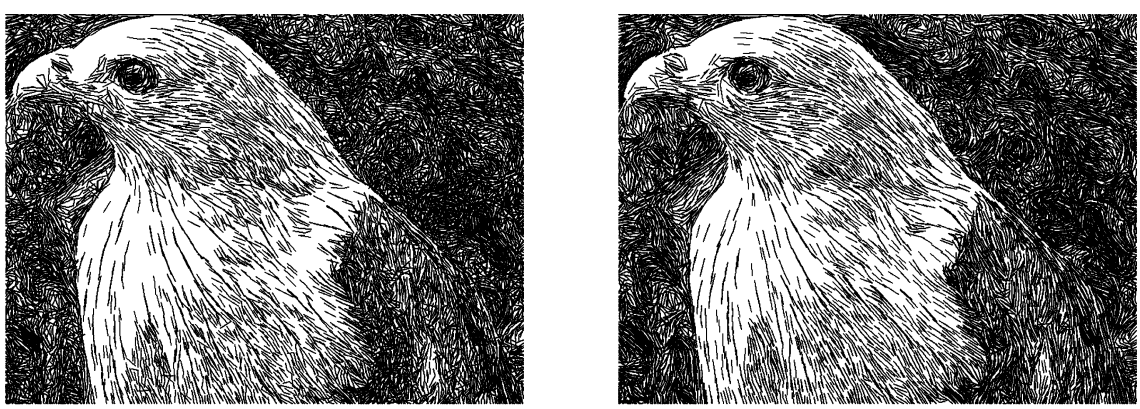


Figure 6.6: Two line drawings with $l = 10$. Left: a smoother field; Right: a less smooth field.

Figure 6.6 shows two sample outcomes from pixel clustering. Both demonstrate line art; the right with a smoother underlying vector field looks more conventionally

(a) $l = 1$ (b) $l = 20$

Figure 6.7: Length influence.

appealing than the left with a less smooth vector field. However, the left result is more random and irregular, which may be considered more natural.

Figure 6.7 shows results from varying line length; the left image uses a very short length, only $l = 1$, while the right image uses $l = 20$. Note that Figure 6.6 used a length of 10. With length 1, all lines follow the smoothed field too closely. Smoothing the field is necessary to remove noise but it also deforms sharp corners to round ones. Closely following the smoothed field produces many round corners that are supposed to be sharp. But with length 20, lines are too long to accurately express the figure. For other results in this chapter, we use the intermediate value 10 for line length.

Skeletonization

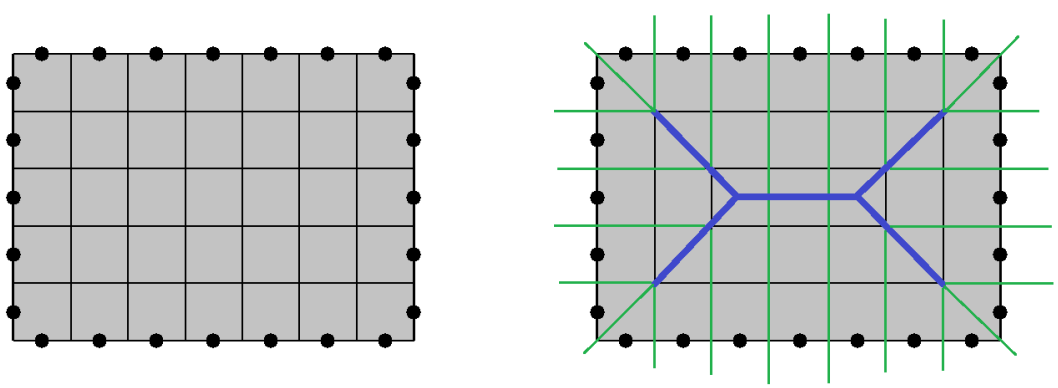
Pixel clustering provides a way to form a long primitive by grouping pixels, resulting in line drawing. However, the long lines are still broken and some features deserve further emphasis. To obtain a clean and large structure of objects, we apply Voronoi-based skeletonization algorithm [36, 130, 162] that finds the medial axis of ETF edges. Medial axis-based methods are highly sensitive to noise, but since we will apply skeletonization only to edges, which contain few pixels and are already narrow and long, the sensitivity to noise is minimized.

We use Voronoi-based skeletonization to obtain skeletons with one-pixel thickness. This skeletonization provides robust results with less sensitivity to boundary noise than skeletonization methods based on distance transform and thinning [166]. The

skeletonization approach works as follows. We obtain an ETF edge image. Each edge is considered a cluster of pixels. Before skeletonization, the system checks the number of pixels in each cluster. We eliminate small edges and isolated pixels. If the number of pixels in a cluster is smaller than a threshold (say 50), this cluster is removed. The value of 50 we used is to avoid short lines. Thresholding the edge image with a large value removes more edges than thresholding with a small value. The remaining edges after thresholding the edge image with a large value would include long lines; but thresholding edges with a small value will retain many small lines. After this removal, Voronoi-based skeletonization proceeds to find the Voronoi medial axis based on the Voronoi diagram of boundary points.

Figure 6.8 (a) shows how to find the skeleton for a rectangle. Each grid cell is a pixel of the rectangular region and the black dots are the boundary points. Figure 6.8 (b) shows the skeleton found by the Voronoi-based skeletonization. The skeleton, marked in blue, is a subgraph of the Voronoi diagram, marked in green.

Figure 6.9 provides an example of skeletonization of simple clusters of pixels. Noisy boundaries do not affect the skeletons. The final skeletons are single-pixel edges and describe the large-scale structure of the image. Two skeleton examples are shown in the second row of Figure 6.10. As you can see, the skeletons show the outlines and the shapes of the bird and the cat, indicating a large-scale structure, but cannot express good tone. The eyes of both the cat and the bird, which are supposed to be black, look very strange and empty because of the loss of tone.



(a) A rectangle and boundary points

(b) Skeleton, marked as blue lines

Figure 6.8: Voronoi-based skeletonization of a rectangle.

(a) Original image (b) Skeletons

Figure 6.9: Skeletons from Voronoi-based skeletonization for simple clusters of pixels.

We adapted the techniques of pixel clustering and skeletonization to create three different effects: simplified line drawing, emphasized line drawing, and combination of stipples and lines. The three different line drawings looks appealing because of long lines and the clean large-scale structure.

6.3.2 Simplified Line Drawing

The ETF skeletons, shown in Figure 6.10, preserve very long lines well, but fail to emphasize strong features and do not indicate tone. To bridge the gap, we suggest adding extra lines to provide structural emphasis and tone.

The strategy arises from observation of artists' work, for example in Figure 2.4. Artists sketched major silhouettes of a subject and also added emphasis or tone along or near existing lines. Shading is then refined over wider areas. To obtain a simplified drawing, our algorithm starts by creating skeletons and then adds lines for tone emphasis. Not all pixels in an image will receive error. First, we apply pixel clustering to the pixels on the skeletons. Error from the set of clustered pixels is diffused to the neighboring pixels, which exclude the pixels on skeletons. After applying error diffusion to the pixels on the skeletons, we then process the pixels within a range of 50 using pixel clustering. Two factors, the magnitude of the vector field and the average intensity of those close pixels, determine whether we draw a

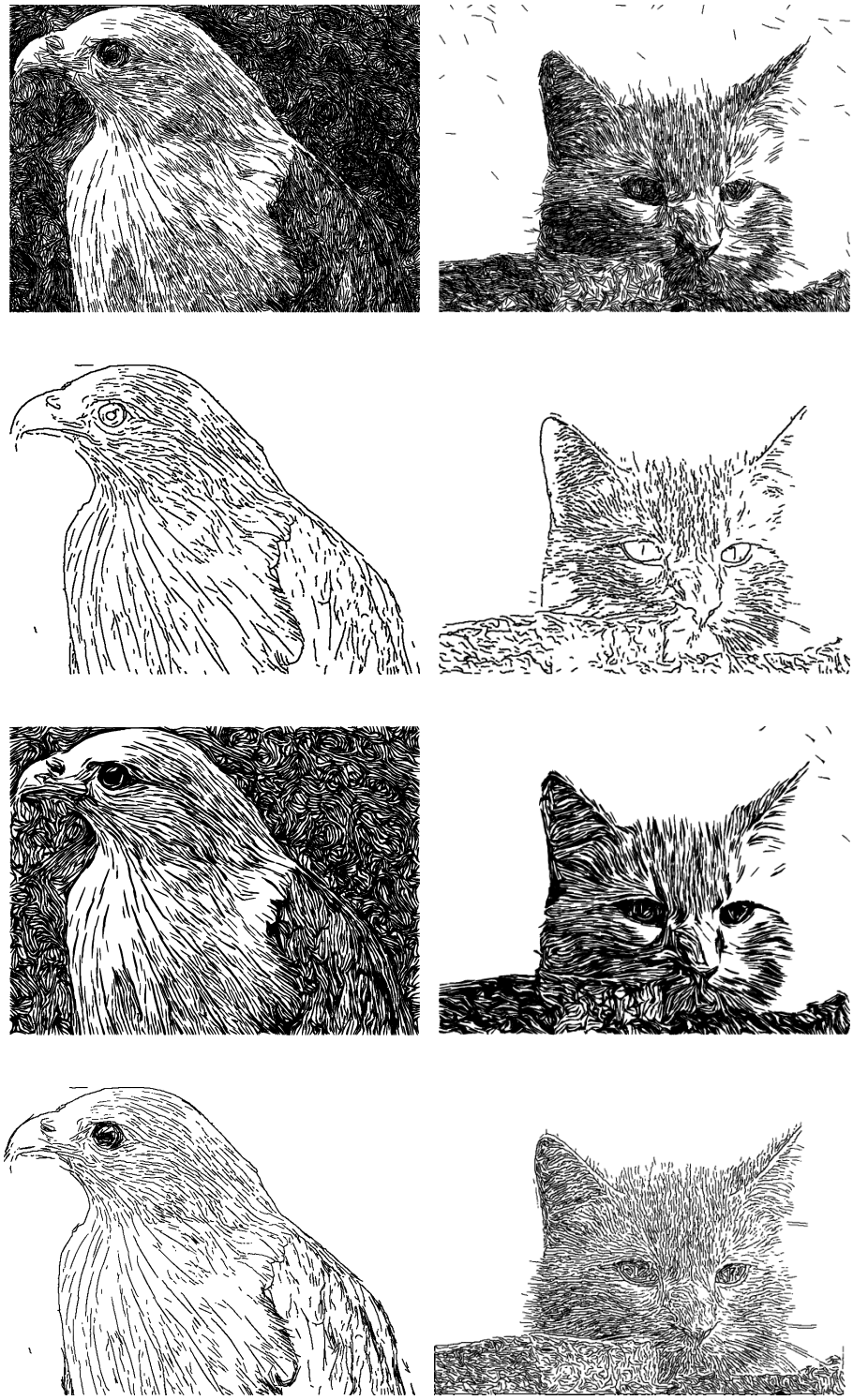


Figure 6.10: Different results using pixel clustering and skeletonization. Skeletonization uses a threshold of 3 to trim edges and uses a factorization value of 1 to obtain fast calculation. From top to bottom: pixel clustering; skeletonization; emphasized line drawing; simplified line drawing.

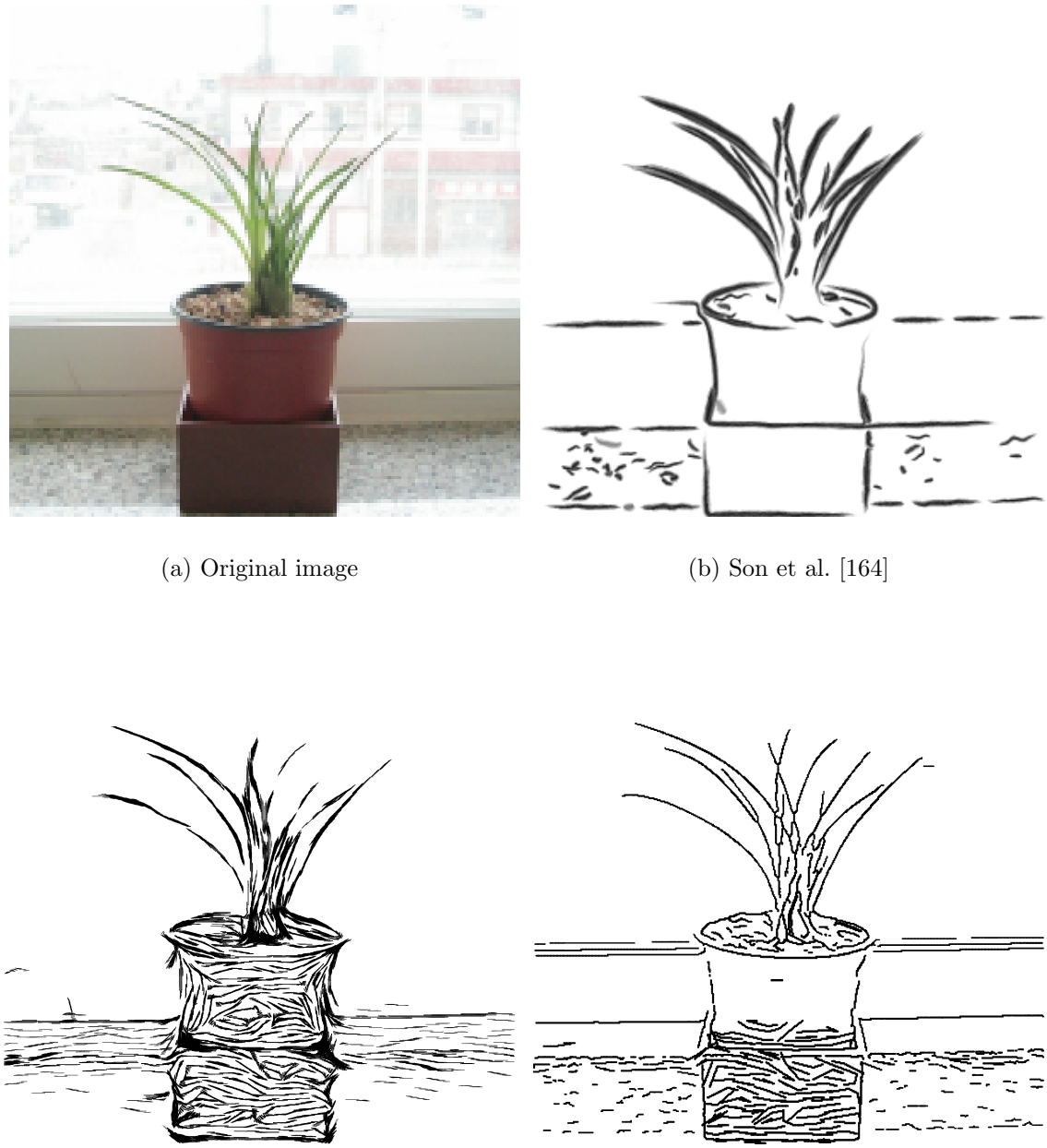
trail or not. If the magnitude of the vector field is smaller than a threshold (say 0.1) and the average intensity is larger than a threshold (say 5), indicating that this is a uniform region and not very dark, lines should not be placed. A very dark area close to skeletons or an area with high magnitude of the vector field is going to be drawn. In order to increase the number of lines used for compensating tone, we suggest a large range of influence distance, a low intensity threshold value, and a large magnitude threshold. To reduce the number of lines, we can employ a small distance range, a high intensity threshold, and a low magnitude threshold. Since simplified line drawings employ very few lines, the overall image becomes very light. We can add lines to very dark areas to recover some tone.

The last row of Figure 6.10 shows simplified line drawings for a bird and a cat. There are few lines apart from the skeleton, but the drawings accurately express the eyes with more lines and give a rich impression of the tone. We compare our results with an abstracted line drawing by Son et al. [164] in Figure 6.11. Particularly notice the box under the plant pot and the soil around the plant. The simplified method supplies tone indication, which in Son et al.'s result is completely missing.

6.3.3 Varied Effects by Pixel Clustering and Skeletonization

I: Emphasized Line Drawing. Artists sometimes emphasize silhouettes or edges. We modify simplified drawing method to create emphasized line drawing. Unlike the simplified drawing method that processes a portion of the pixels, the method for emphasized drawing deals with all pixels. The lines are formed by pixel clustering. The trail extension process halts not only when a trail increment is white, but also when an increment reaches part of the skeletons. The pixels on the skeletons will not receive error; other pixels will receive error and change their intensities. Thus, the pixels on the skeletons will maintain their initial tendencies to be black or white. For the skeletons in a dark region, a pixel maintaining initial tendencies is likely to be black. In this way, there will be more black pixels clustered to form lines on the skeletons, which makes the lines seem thickened and emphasized. The third row of Figure 6.10 shows line drawings with thickened edges.

Skeletons are a thinned version of the thresholded ETF edges; the thresholding removes isolated and small edges and cleans the structure of objects. Thickening the



(a) Original image

(b) Son et al. [164]

(c) Emphasized line art

(d) Simplified line drawing

Figure 6.11: Comparison with line drawings.

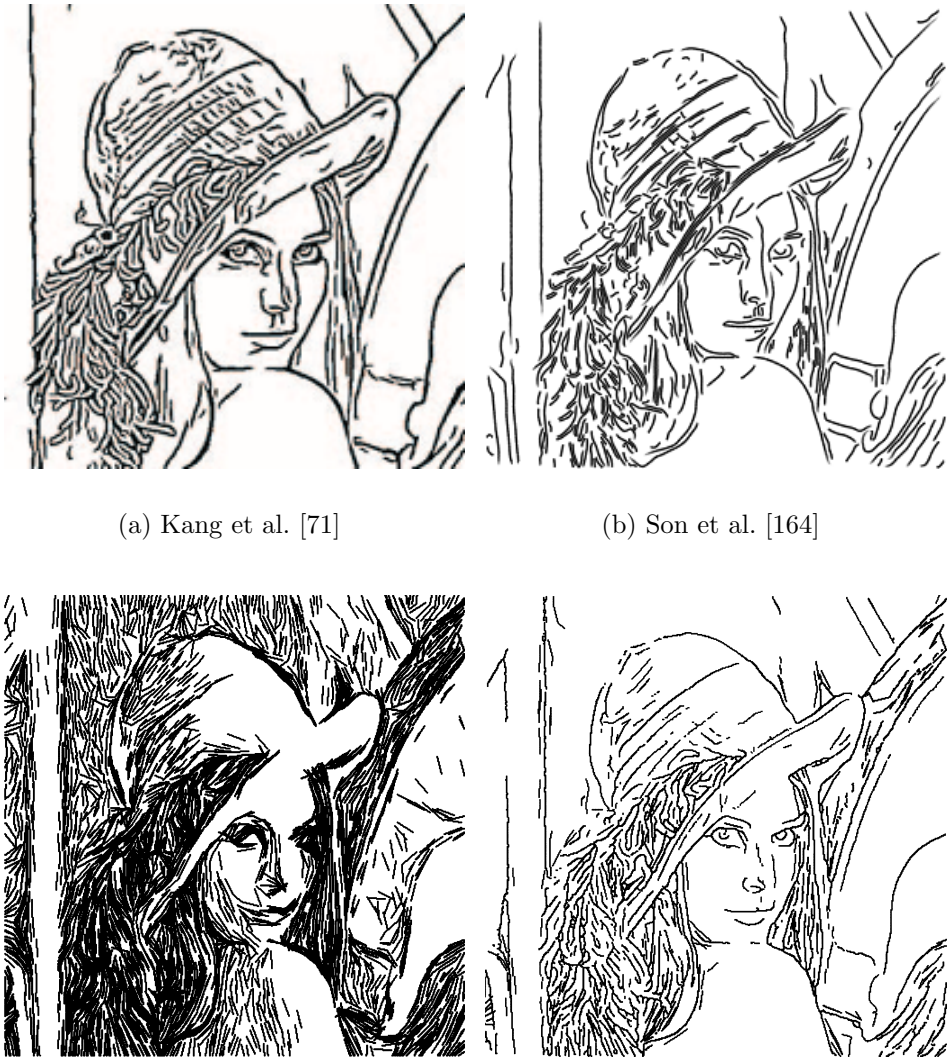
skeletons that represent the major structure can enhance the features visually. The emphasis of eyes and hair makes the bird and the cat attractive, particularly because the strong textural features are represented by the lines very well.

Figure 6.12: Both use $P(5, 5, 0, 10)$. Left: stippling; right: stipples and skeletonization.

II: Combination of Stipples and Skeletonization. Line drawing and stippling often coexist, both in pen-and-ink illustration and in media such as scratchboard. Hedcut stippling, presented by Son et al. [164], is a related example. With the earlier contrast-aware and priority-based scheme for stippling, we can adjust parameters to control the contrast by presenting a distribution with high difference in dot densities, which emphasizes important features such as edges. However, for linear features, lines are a clearer representation than dots. Therefore, we suggest using both lines (obtained from ETF skeletonization) and stipples in the same process. The algorithm begins by dealing with the pixels on the skeleton, processing them one by one and applying error diffusion; the error diffusion does not change the intensities of skeleton pixels. The remaining pixels are then processed using the SPS approach, with the resulting dots mainly responsible for tone indication. Figure 6.12 shows a result having a combination from stippling and lines. The house architecture in the output image is nicely shown by lines while the tone has been taken care of by the stipples. Compared with the basic stippling result on the left, this combination displays the large-scale structure more clearly. Particularly note the outlines of the windows, which are only approximately shown in the stippled result.

6.4 Results and Discussion

In this chapter, we briefly described the conversion from stippling to line drawing with short lines. The focus then turned to creating simplified line drawings. The simplified



(a) Kang et al. [71]

(b) Son et al. [164]

(c) Emphasized line art

(d) Simplified line drawing

Figure 6.13: Comparison with previous line art.

line style employs pixel clustering to obtain long lines. Skeletonization produces a large-scale structure to promote line art nicely. The techniques of pixel clustering and skeletonization can also generate emphasized line drawings and the combination of stipples and lines. An emphasized line drawing enhances the important edges as well as indicating tone. In the combination of stipples and lines, important features are expressed by skeletons and stipples take care of tone.

Here, we present comparisons with other automatic image-space line drawing methods. Figure 6.13 shows a comparison with previous line art rendered by Kang

et al. [71] and Son et al. [164]. The simplified result displays Lena's eyes and hair more neatly than Kang et al.'s and Son et al.'s result because of the use of long lines. Furthermore, the simplified result uses additional lines to add tone; particularly note the hair area and the frame of the mirror. Tone is not displayed in Kang et al.'s and Son et al.'s results at all. Our emphasized result uses heavy lines as Kang et al. and Son et al. did. The short lines for Lena's eyes and hair can indicate tone, which is totally lost in their results. Subjectively, we prefer simplified line drawings to emphasized line drawings because simplified line drawings look clearer. The emphasized result looks very dark around the strong edges since the emphasized method did not reduce the number of lines.

Figure 6.14 shows another comparison with previous line art. The result by Winnemöller et al. [188] uses pencil texture and hatching to add shading on the face to represent a hatching style. Our simplified line drawing not only expresses the man's hat with clean and long lines, but also emphasizes his eyes and hair with more tone indication, which is not shown in their result. The emphasized line drawing in this example does not look good due to many small lines, darkening the regions and then missing the details. The eyes and mouth are not nicely displayed either.

The first image in Figure 6.15 shows a real drawing by Belzen. Belzen only sketched the silhouettes of the woman's face, her nose, her lips, and her hair outline. Except for the emphasis on the eyes by darkening, we can barely find tone change and there is no texture information about her face and her clothing. Our simplified drawings are faithful to this clean line style. The long lines represent the hairy sheep, the complicated building, and the delicate woman's face gracefully; the dark areas such as the chest of the sheep, the building roof, and the eyes of the sheep and the woman have been darkened with numerous small lines. However, this is far from the work by an artist. Although the line drawings we obtained are quite good, the lines are isolated without clues about which objects they depict. Further exploration should focus on connecting line properties such as thickness to the semantics, which can be done manually.

The lines displayed in this chapter are groups of dots, which are located in a grid and are continuous spatially. The continuity in the grid forms lines. To improve the representation of lines and reduce the number of primitives used, one improvement is

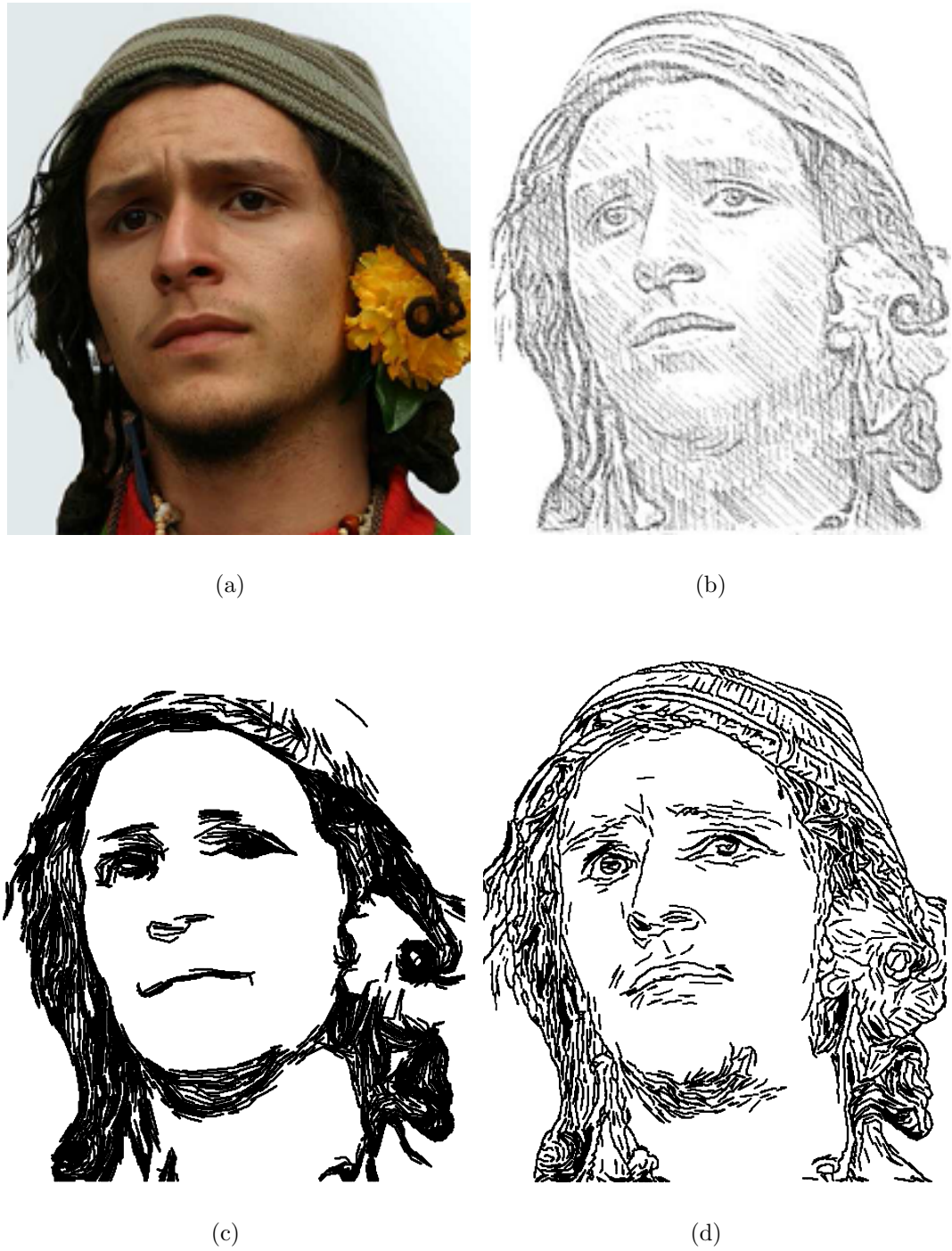


Figure 6.14: Comparison with previous line art. (a) original image; (b) Winnemöller et al. [188]; (c) emphasized line art; (d) simplified line drawing.

to generate a curve for each group of dots by interpolation or curve fitting methods.

This chapter visually compares line art with results by other researchers. We

Figure 6.15: First: a drawing by René van Belzen. Others: our simplified line drawings.

Method	Image	Size	Time (seconds)
Emphasized line drawing	Figure 6.10 (e)	828× 621	42.20
Emphasized line drawing	Figure 6.10 (f)	828× 621	56.47
Simplified line drawing	Figure 6.10 (g)	828× 621	37.68
Simplified line drawing	Figure 6.10 (h)	828× 621	36.19
Emphasized line drawing	Figure 6.11 (c)	480×480	17.61
Simplified line drawing	Figure 6.11 (d)	480×480	4.82
Emphasized line drawing	Figure 6.14 (c)	459×530	31.3
Simplified line drawing	Figure 6.14 (d)	459×530	10.63

Table 6.1: Processing time on an Intel(R) Core(TM) i3-2350M CPU @2.3GHz with 8GB RAM .

also apply our methods of line drawing on many images. All drawings with short strokes from our line art method can preserve details in highly-textured regions. An explanation is that stipplings from SPS place stipples along edges and the lines following the vector field enhance the edges further. Simplified line drawings are good too, but there are some unattractive results from our method for emphasized line drawings, such as Figure 6.10 and 6.14. The emphasized line drawings fill in all the space based on the background vector field and the important edges are enhanced by thickening the lines. However, some regions have very low magnitudes in the field. For example, the background area in Figure 6.10 is almost uniform, barely showing directional features. Pixel clustering and trail extension handle those smooth regions poorly and long strokes are not easy to obtain in those regions. One possible improvement is to design a fake direction field for those regions. This shows our limitation for effects using only black or white. Drawing lines with only black or white, would make images very dark or very light, which would wash out the image (e.g., eyes in both Figure 6.13 (c) and 6.14 (c)). Introduction of more colors such as greyscale could improve the visual quality.

Table 6.1 provides the processing time for our results. Figure 6.10 (e) and (f) are emphasized line drawings and Figure 6.10 (g) and (h) are simplified line drawings. It took a longer time for our method to produce Figure 6.10 (e) and (f) than to produce Figure 6.10 (g) and (h). The reason is because the method for an emphasized line drawing applies error diffusion to all pixels and in generating a simplified drawing, error is only diffused to the pixels on and around the skeletons. The emphasized

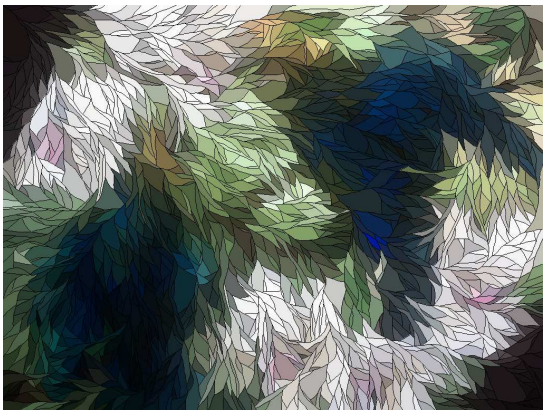
line drawing needs more time because error diffusion acts on more pixels. Times for computing simplified line drawings vary depending on the number of pixels in the skeleton.

6.5 Summary

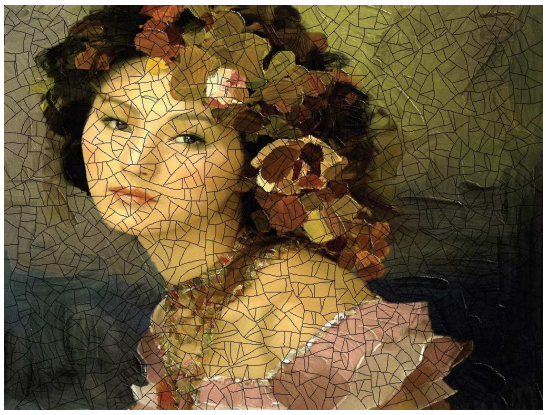
This chapter extends SPS to line drawings. We focus on creating long lines, which are generated by pixel clustering and skeletonization. The key is to group a set of pixels when using error diffusion and meanwhile to include large-scale skeletons from ETF edges in this simplified style. Simplified drawings display clean and elegant results. At the same time, the techniques used for simplified drawings also can create emphasized line drawings and a style with stipples and skeletonization.

Chapter 7

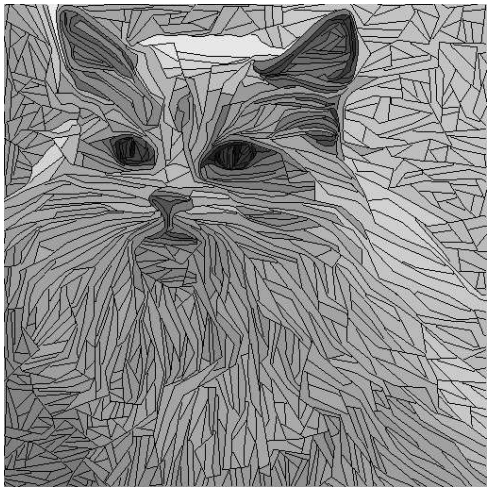
Artistic Tessellation



(a) Tiffany glass



(b) Craquelure



(c) Mosaics

(d) Macaroni art

Figure 7.1: Artistic tessellations by growing curves. (a) Mosaics; (b) natural patterns; (c) mosaics; (d) Macaroni art.

7.1 Introduction

In this chapter, we describe a method to artistically tessellate a region by growing curves. The idea of artistic tessellation (AT) by placing curves was inspired, by accident, when we did experiments on tessellation by drawing curves on a paper manually. We found the placement of curves can have the ability to demonstrate abstract and natural patterns such as bricks, or cracks. For example, if we randomly cut a region into small pieces in vertical and horizontal directions, the final partition is made with a set of small rectangles. The small rectangular tiles remind us of bricks. For another example, if we tessellate the region in a random way, such as in a random direction or using irregular cutting lines, the final small irregular tiles can remind us of cracks. Furthermore, when we place the curves, there is a choice of location, direction, length, and the duration of growth. We can terminate a curve when it meets a previously placed curve. It looks as if the later curve is occluded by the earlier curve, suggesting that the earlier curve is closer to us than the later one. The occlusion can suggest the spatial relationships between objects and produce an illusion of 3D. All of those observations inspired us to investigate tessellation by placing curves.

We implement the tessellation method using a particle system, which flexibly provides good control over the final results by variations of the initial placement, the placement order, the starting direction, and the dynamics. We also propose an automatic image-based mosaic method using a smoothed vector field to guide particle movement. The final irregular tessellation simulates stained-glass mosaics where elongated curved tiles suggest the content of highly textured areas. Additionally, based on the basic idea of artistic tessellation, we introduce splitting and stacking techniques. Splitting is a technique to partition a region step by step using curves; stacking is a technique to align a set of curves along a given growth path. We show some applications employing splitting and stacking techniques to create illusions of 3D shapes, abstract patterns, and natural patterns, such as cracking and scales.

Contributions. We investigate the placement of curves in generating tessellations. The partition from the growth of curves can create irregular and natural patterns. We create stained-glass mosaics using the placement of curves, which are

guided by images. The proposed splitting and stacking techniques can generate natural and abstract patterns including cracks, scales, and rivers. Figure 7.1 demonstrates applications of our artistic tessellation technique. In NPR research, there is a clear division between region-based methods and stroke-based methods. The idea of using a set of curves to obtain a partition brings a way to convert curve primitives into a region primitive. The illusion of 3D shapes from 2D curve distribution is a new topic in NPR.

7.2 Tessellation by Growing Curves

We strive to create tessellations with good visual appearance. Wong et al. [191] stated that repetition, balance, and conformance to geometric constraints are three elements of artistic design. Tarbell [173] generated patterned effects by placing curves on the plane in order to present city-like structures. Our tessellation algorithm is also based on placement of curves for pattern generation. We propose to present similarity in partition by providing similar control over curve generation and maintaining spatial constraints to fill the space. A tessellation could be constructed by placing curves in a region. The curves form the tile boundaries of the partition. Our method for artistic tessellation is described as follows. Given a blank region and a direction field, our tessellation begins with a starting distribution. Curves grow from points sequentially or in parallel. For each curve, its growth stops by reaching either another curve or the image boundary. Very short curves will be removed to prevent narrow spacing. If a curve passes too close to previous curves, the curve is removed as well. Variations on the initial starting distribution, the order of the growth, the orientation of the growth, and the properties of the curves (such as curvature, length, and thickness) control the final appearance of the partition. One of our intentions is to create natural patterns, which appear irregular and random in the partition. We propose some rules to build artistic tessellation, described as follows.

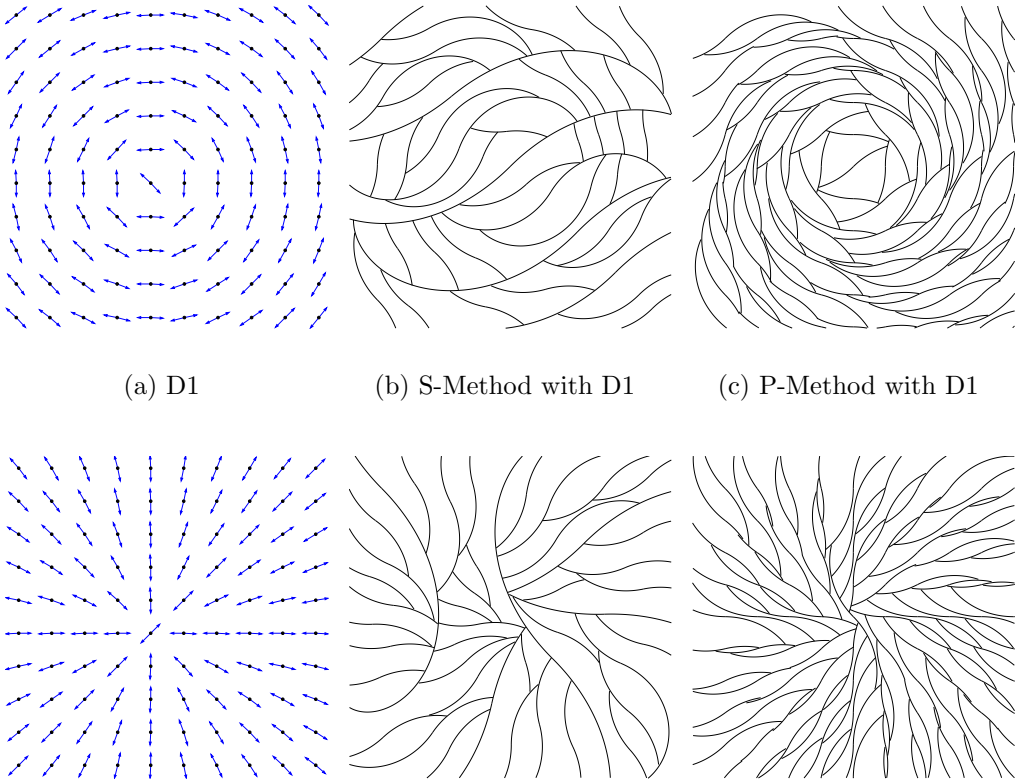
- When we generate curves, similar properties should be used along the growth of curves, such as curvature and the change of curvature; the growth should start with similar orientations also. The enforcement of the similarities in curve generation can produce regions with similar shapes that provide a sense of unity.

- We do not allow short curves, and narrow spacing between two curves is also forbidden.
- We do not allow two starting points to be placed too close together.

The introduction of irregularities makes each partition appear natural. Irregularities are obtained by varying parameters of the curve properties and the curve starting orientation; further obvious randomness derives from the unexpected intersection locations of curves. The process of curve growth can be either sequential or parallel. In the sequential case (S-Method), the system starts growing at one point and processes curves one by one: a new curve has to wait until the previous one is completed. In the parallel case (P-Method), the system begins with a set of points, initializes multiple curves simultaneously, and then grows them incrementally in parallel.

Examples from both the sequential method and the parallel method appear in Figure 7.2. Notice that the tiles in those tessellations are irregular curved and elongated shapes. Similarities occur in the whole tessellation because the boundaries of tiles present similar curvature and similar starting angles show repeatedly. The similarity comes from the same curve generation process. However, S-Method and P-Method create different appearance because of the order of operations. The parallel method preserves the large-scale structure, showing in the initial distribution, better than the sequential method. In general, by using P-Method, curves are shorter and hence have little time to vary much from their initial direction. There is not much room to have spatial control. Conversely, the sequential method grows curves one by one. S-Method can adjust the placement for each curve individually and apply the spatial control for each one, which is flexible, taking into account all previous curves. Doing the placement sequentially often makes the order of growth apparent, which may be undesirable, although for some patterns, such as cracking, S-Method is useful.

Figure 7.3 illustrates the process for the S-Method. The system begins at a random point and grows the first curve in two opposite directions. Next, a distance map stores the distance of all locations to the nearest point on either a curve or the region boundary. The second curve grows at the point of maximum distance, avoiding narrow spacing due to close placement. The process updates the distance map and iterates the growth of the next curves repeatedly. The growth of the partition stops when the maximum distance value is below a threshold.



(a) D1 (b) S-Method with D1 (c) P-Method with D1

(d) D2 (e) S-Method with D2 (f) P-Method with D2

Figure 7.2: Four examples from the basic idea. Left: Two direction assignments (D1 and D2); middle: the sequential method; right: the parallel method.

We show the process for P-Method in Figure 7.4. P-Method starts with a distribution of points. For example, Figure 7.4 begins with a 5×9 grid of points. At the beginning, all curves are grown simultaneously in fixed timesteps. Some curves will stop their growth when they intersect with other curves or the region boundary. Other curves will keep growing if there are no intersections with the existing curves and the boundary. The process ends when all curves have stopped.

In principle, any curve generator fits for this strategy. We propose to use a particle system [145] with forward Euler integration. For each time step ($\Delta t = 0.01$), each particle moves to a new position x based on previous velocity v_0 and previous position x_0 calculated from the dynamics. The time step Δt is a small value to allow a small movement each time. A curve is created from the sequence of positions of a particle. The key computations for dynamics are as follows: $a = F/m$ for a force F, $v =$

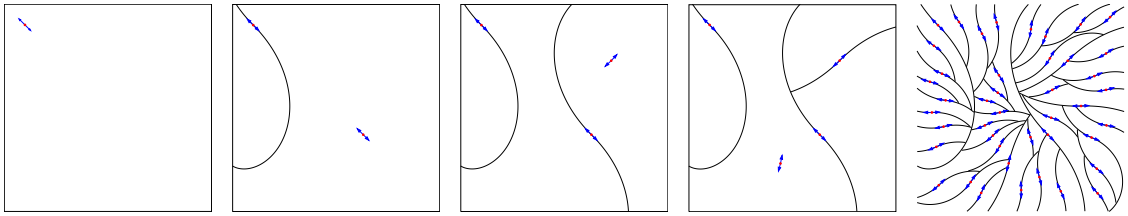


Figure 7.3: Progression of S-Method (D2 vector field).

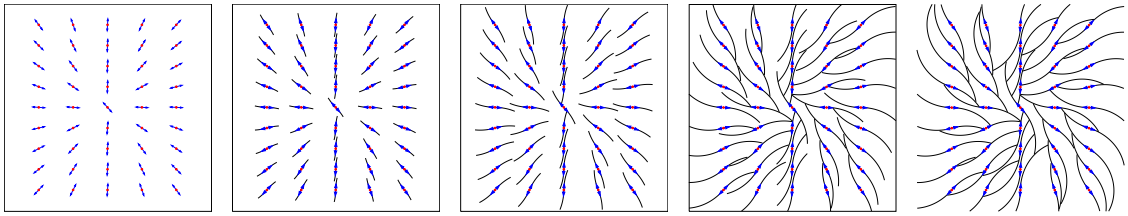
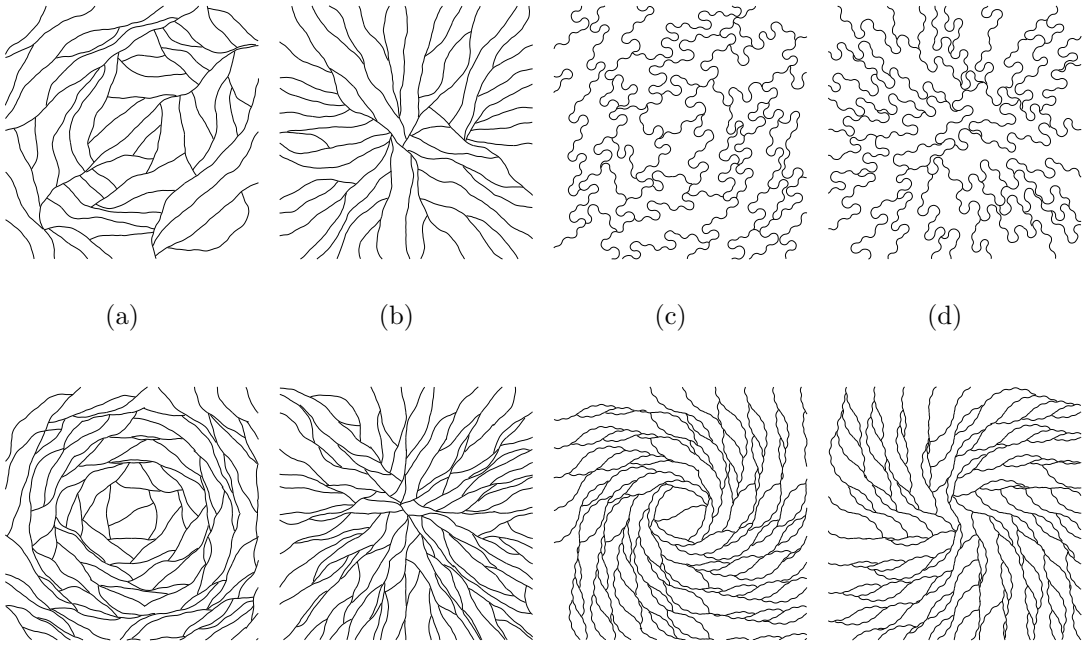


Figure 7.4: Progression of P-Method (D2 vector field).

$v_0 + a \times \Delta t$, $x = x_0 + v \times \Delta t$, and unit mass $m = 1$. We intend to use the particle system to create patterns and mosaics. We use different force configurations for different purposes. For mosaic generation, the system reads $\overrightarrow{F}$ from a vector field, guided by image content; for other patterns, we use the Lorentz force in a magnetic field, previously used to generate magnetic curves [194]. Figures 7.3 and 7.4 were generated using magnetic curves; the following sections will present details. Given the same initial configurations for starting locations, direction field, and starting directions, Figure 7.5 shows curve variations for creating another group of tessellations. The partitions not only globally maintain the same directional impressions as Figure 7.2, but also contain further small-scale details of curve properties.

7.3 Texture Indication for Mosaics

A vector field calculated from an image can indicate the image content. A tessellation can be created guided by an image including color and edge. We can simply use the existing tessellations from the previous section and assign the average color, calculated from image color, to each tile. Some results are shown in Figure 7.6. In these cases, the tessellations are unrelated to image edges. The curves do not depict any objects. The average color for each tile helps indicate color information from the original lotus

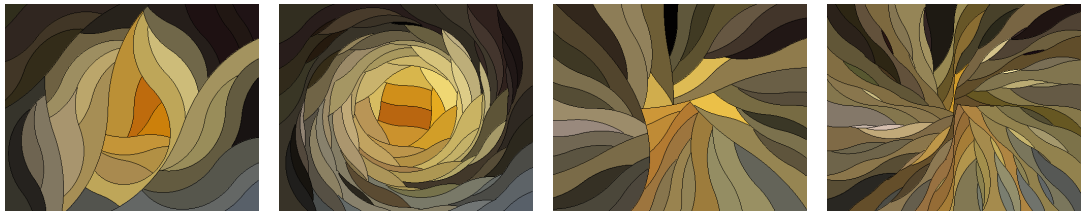


(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 7.5: Curve variations. Top: S-Method; Bottom: P-Method.

image, shown in Figure 7.7, but the lotus itself cannot be seen.



(a) (b) (c) (d)

Figure 7.6: Abstraction from Figure 7.2. (a) S-Method with D1; (b) P-Method with D1; (c) S-Method with D2; (d) P-Method with D2.

We would like to present representational mosaics, and especially to indicate texture using flow. Our particle system uses the edge tangent field of an image to create the force field. The force enforces a particle to trace the vector field. If a new particle becomes too close to a previous curve, this particle will be removed. We use s_{min} to represent the minimum distance.

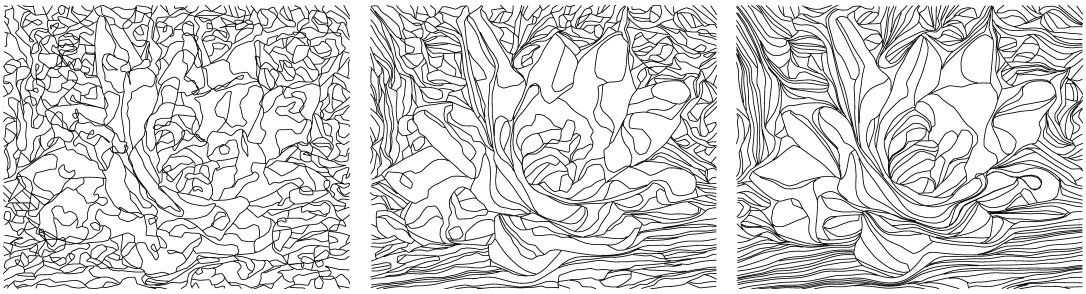
If the starting position is not at the region boundary, the particle system seeds

Figure 7.7: A lotus and its vector field.

two particles at each location. Both two particles read the force from the vector field. The force direction of particles is opposite to each other. One particle uses $\overrightarrow{F}$ and the other particle uses $-\overrightarrow{F}$. Both particles initialize their velocity to zero and start tracing along the vector field under the forces. We use the edge tangent flow (ETF) [71] to calculate the vector field, in which the vectors are perpendicular to the image gradient. Possible initial point locations come from a set of points created by the structure-based stippling method by Mould [123]. Particles update their position and velocity with forward Euler interpolation and proceed in discrete timesteps. The current velocity of each particle is an integrated value after running several timesteps. Initially, a particle under a constant force traces the field in a straight direction. The trail of the particle is a straight line. After a few timesteps, the velocity direction is the same as the initial force. However, since the vector field represents the content over the space, the current force read from the vector field may have a different direction. To have a meaningful tessellation faithful to the image, each particle should closely trace the vector field, which means the force used in tracing has to be changed regularly. We use A to determine when the force should be updated and read from the vector field. If A is 4, it means that the particle moves 400 timesteps and our system then reads a new force. We suggest to use $A > 2$ to make sure that a particle moves at least 200 timesteps. If A is a small value, the timesteps are small. The force is updated quickly, which creates a short increment and produces a velocity direction too close to the field. Tracing the field too closely does not produce a good result. The field is affected by many factors such as lighting condition and object surfaces. In addition, how and when the image was taken affect the field. Basically, the field is noisy. The noise in the field would change the direction

of the force at a pixel, which diverts from the image content, sometimes becoming completely unrelated to the image. If we did not address the problem of too close tracing, the curves would faithfully follow the noise in the field and would describe the image content hardly. We introduce A to avoid too close tracing. A particle moves under an old force for a number of timesteps, creating a straight line. The force is only updated when the timer runs out. However, a large A value (e.g., greater than 8) causes the curve to be a very long straight line, unable to display much image content. A discrete map storing the discrete locations of particles' trails is used to detect collisions between particles and curves. A particle stops tracing when its trail collides with an old curve or the region boundary. Tracing is also stopped when one of the eight neighbors of the particle's position has been previously traced by another particle.

The direction of a field can be the direction of the force or the opposite direction of the force. We inspect the dot product of the current and previous force vectors to determine the direction of the force used in Euler integration. $\overrightarrow{F}_1$ means the current force and $\overrightarrow{F}_0$ means the previous force. The system checks the value of $\overrightarrow{F}_0 \bullet \overrightarrow{F}_1$. If $\overrightarrow{F}_0 \bullet \overrightarrow{F}_1 < 0$, the direction of $\overrightarrow{F}_1$ is flipped. The flipping prevents the curves from doubling back on themselves suddenly. We smooth the field in order to remove noise. The vector field for a textured region contains unexpected directions due to noise. Smoothing removes noise by averaging the neighboring directions. The averaging can strengthen the main trend in textured areas. We employ Kang et al.'s bilateral approach [71] to smooth the field. We smooth using a 9×9 window. Particles moving on a smoothed field produces smoother curves. Visually, smoothed curves lack doubling or jaggedness. Figure 7.8 shows how smoothing affects the final tessellations. Figure 7.8 (a) was generated without any smoothing so the curves change direction and curvature quickly and look very noisy. Progressively smoother curves appear from (a) to (c); we consider the smoother curves, and corresponding tessellations, to be more aesthetic. Smoothing the field over 10 times causes very smoothed curves, which lose many sharp features. The sharp features display the major details of an image. Since we would like our abstracted style to preserve information originally shown in the image as much as possible, we suggest to use values between 2 to 10 to have a good tradeoff between the preservation of content and the smoothness of the



(a) Unsmoothed (b) Smoothed twice (c) Smoothed 5 times

Figure 7.8: Difference in smoothing ($A = 4$).

curves.

(a) (b)

Figure 7.9: (a) Our line art with $A = 8$ and smoothing 5 times; (b) Son et al.'s line drawing.

The variable A is a parameter to indicate when to update the force in forward Euler integration to the current force read from the field. The adjustment to A can control the degree of faithfulness to the image content. The larger the A, the more the particle diverts from the original field. Figure 7.9 (a) uses straight lines to draw a flower, showing this diverted result. Compared with Figure 7.8 (c), the trails are not faithful to the field but the underlying texture is still implied. Figure 7.8 (c) and Figure 7.9 (a) can be considered drawings with smooth curves and straight lines; Son et al. [164] created their line drawing by using a similar vector field, shown in Figure 7.9 (b). The drawings we created fill the entire space and indicate texture. The flower can be seen clearly, although we did not add colors.

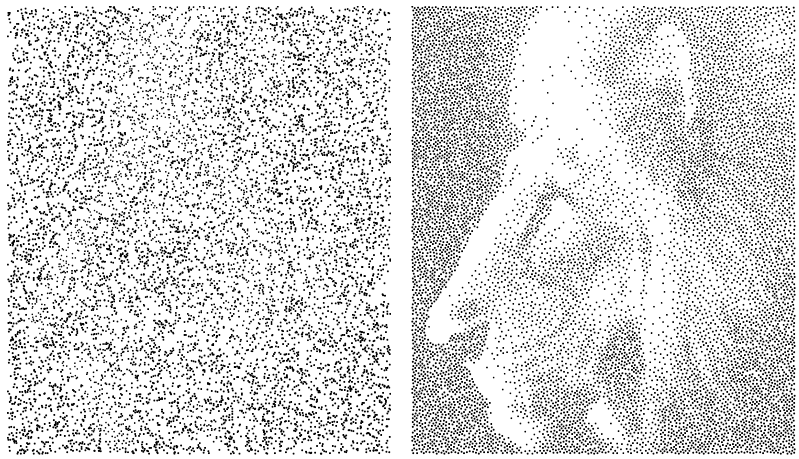


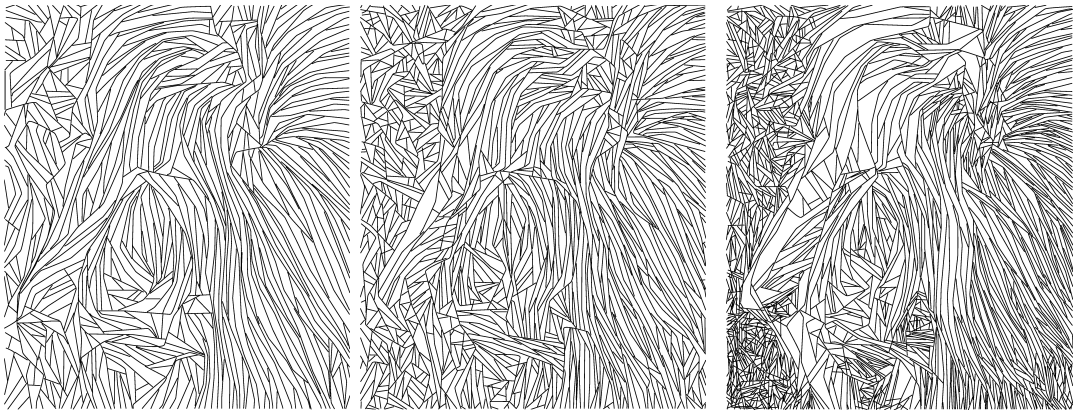
Figure 7.10: A random distribution and a better distribution from Mould's stippling.

The initial distributions influence final results. Under the same stipple count, in Figure 7.10, we compare the result starting from a random distribution of stipples with the result from a better distribution created by structure-aware stippling by Mould [123]. Figure 7.11 (b) and (c) look better than the one in (a) without consideration of structure in initial distribution; Figure 7.11 (c) is further improved with intensity indication by using Equation 7.1 that adjusts the minimum spatial distance,

$$s_{min} = dist_{min} + DIST \times (I/255) \tag{7.1}$$

where I is the intensity value of a point. Both $dist_{min}$ and $DIST$ in the equation are constant values, indicating the minimum spacing and the offset respectively. If we choose larger values than $dist_{min} = 2$ and $DIST = 5$ in the equation, s_{min} will be larger. The minimal spacing to place curves gets bigger. The tessellation will be sparse and use fewer curves. Conversely, if we choose smaller values than $dist_{min} = 2$ and $DIST = 5$ in the equation, the partition will be denser.

Figure 7.12 displays curves and colors each tile with the average color from the original image. Both results nicely show the feather texture for the eagle and the furry texture for the lion. The elongated curved tiles resemble Tiffany glass. Tiles used in stained-glass mosaics are much smaller than those in Tiffany glass. Big tiles can be further subdivided by adding new particles. If the arc length of a curve exceeds a threshold, the system emits new particles along the curve. We use 50 as the length threshold. A larger threshold will produce larger tiles. The order of generation is



(a) (b) (c)

Figure 7.11: The difference in distribution with the same stipple count. $A = 8$ with smoothing 5 times. (a) Uniform random distribution; (b) structure-aware stippling without intensity indication; (c) structure-aware stippling with intensity indication.

from the start of the curve onward, which maintains the spatial characteristics; the child particles use force directions perpendicular to their parent force directions. The parent particles use the directions read from edge tangent field as force directions. The new particles rotate the tracing by 90 degrees and alternate the sign of their initial directions. The new trails are grown on both sides of the parent curve, thus subdividing long tiles into small tiles. Figure 7.13 shows how our partitions represent two highly-textured examples. This figure reminds us of the rectangular brush strokes and the tessellations shown by Haeberli [47]. Haeberli explicitly assigned the stroke directions according to the original image. Each position is an isolated case without looking at neighboring positions. Our method traces the vector field and smoothes noise. The movement of a particle is affected by the vector field and the existing curves. The termination and the starting placement of a particle are determined by the surroundings too. Our curves nicely show the texture.

We also apply artistic tessellation to create macaroni art, an art style typically seen in children's work. Different objects (say, dry pasta) are pasted onto a canvas to form an image. Figure 7.14 shows macaroni examples by aligning 3D cylinders along the curves. We choose POV-Ray to render the cylinders. The resulting images display their constituent curves very prominently.

Figure 7.12: Flow-preserving mosaics based on the S-Method with elongated tiles. $A = 4$, with two smoothing passes.

7.4 Applications to Abstract Patterns and Natural Patterns

Our basic tessellation method can directly create patterns that do not have concrete meaning for objects. Next, we do a further exploration to generate natural patterns, resembling structures found in nature such as cracks, scales, and rivers. We also investigate a tessellation method to generate an arrangement of curves which can display apparent occlusions to produce an illusion of an ambiguous 3D scene. All the images in this section are created by the sequential method. The force is calculated in a magnetic field and particle trails create magnetic curves whose curvature can be adjusted by the charge parameters. The ability to control the curvature gives us many variations in curves. Each particle moves with a charge q in a magnetic field $\overrightarrow{B}$, experiencing the Lorentz force. A magnetic curve is the trail of such a particle.

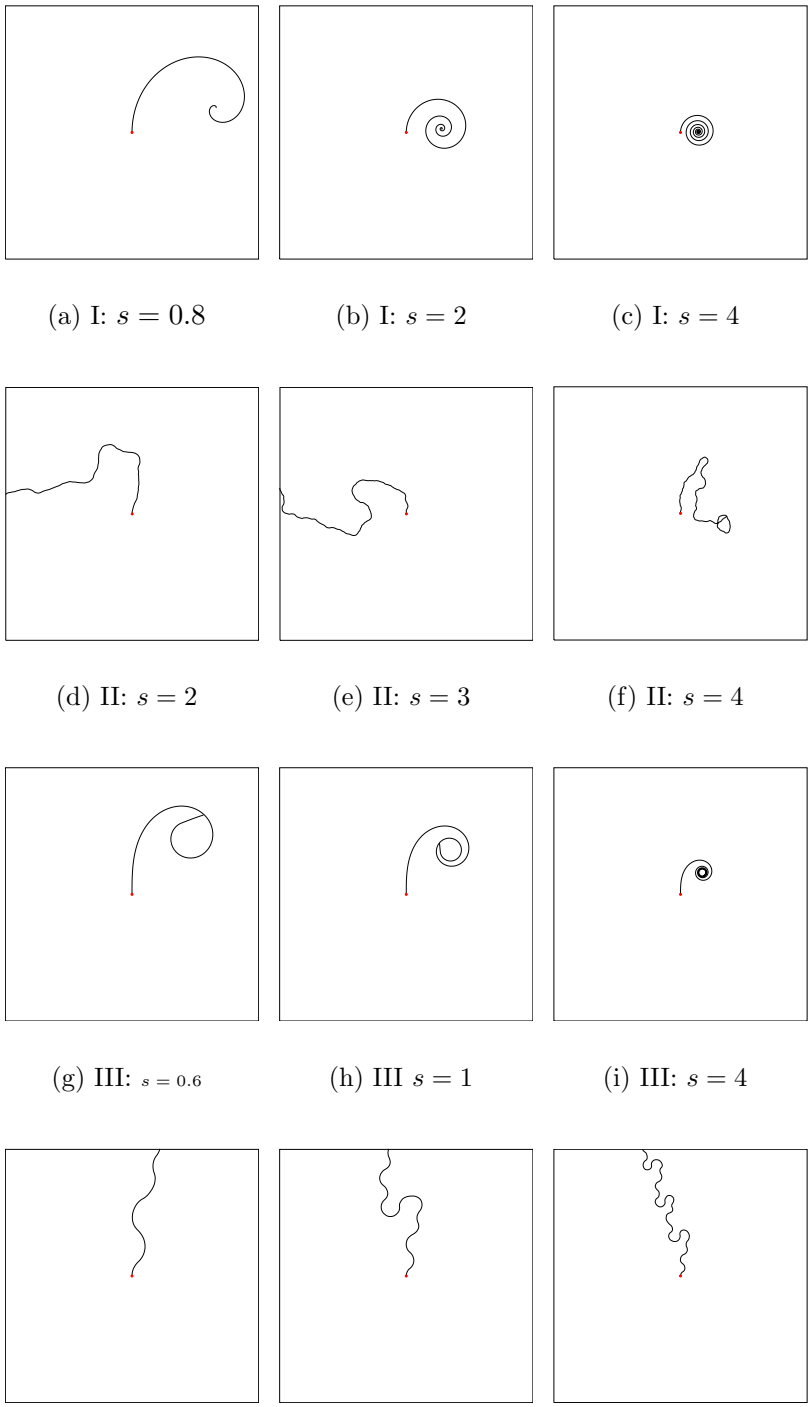
Figure 7.13: Stained-glass mosaics. $A = 4$. Smoothing 5 times.

Figure 7.14: Four examples of macaroni art.

A constant $\overrightarrow{B} = \{0, 0, -1\}$ is used. The velocity $\overrightarrow{v}$ is restricted to the xy plane, as is the curve.

$$\overrightarrow{F} = q\overrightarrow{v} \times \overrightarrow{B} \tag{7.2}$$

Equation 7.2 shows that if $\overrightarrow{B}$ is constant, the magnitude of the charge q affects how a particle moves, implicitly controlling the curvature of the curve. If the sign of a charge is flipped, the curvature is also reversed. The direction of the initial velocity determines where the curve grows. We use $q = s \times f(t)$ to adjust the charge and to control the overall curve shape; s is a parameter to control the curvature magnitude and $f(t)$ is a function to control curvature change. In the following, if we do not specify s explicitly, it means $s = 1$. Figure 7.15 shows some curve examples. Curves in Figure 7.2, which we call curly curves, use $f(t) = (500-t)^{0.8}$, where t is less than 500 in our case. Figures 7.15 (a), (b), and (c) also use curly curves. The output of function $f(t)$ for curly curves reduces with time (t) increasing, where the constant values (500 and 0.8) used in the function $f(t)$ affect how the output values grow. In Equation 7.2, q is reduced with time and as a result the curvature of the curly curve increases. The figures (a), (b), and (c) show how the curves look as the curvature increases. A user can define their own function $f(t)$ to have control over the curvature change with time. The basic tessellation idea works for any curve generation and the results will differ depending based on the curve definition. The random irregular curves shown in Figure 7.15 (d), (e), and (f) are generated by randomly assigning a new value to $f(t)$ in each timestep. The new value has a random sign for the charge and the magnitude of $f(t)$ is in the interval $[0.00001, 0.1]$. The values from this interval range are small so that the curves are generated with small curvature, which reduces the chances of self intersections. If a user chooses large values for the range, the curvature will be very large. If the range has a large span, the curvature will change dramatically and curves will look more irregular. We do not allow the curves to become too varied, because unpredictable growth of a curve is unhelpful to control the partition. If curves are too variable, the curve shapes will control the final effects of the tessellation. The varying curves make the assignment for the path and the particle orientation become trivial. Figure 7.5 (c) and (d) show Jigsaw-like puzzle examples, which employ s-shaped curves with function $f(t) = sgn(P(t)) * (f(t_0) + Random(0.0001, 0.01))$, where $f(t_0)$ is a constant value that determines the amount of curvature initially; $sgn(P(t))$



(a) I: $s = 0.8$ (b) I: $s = 2$ (c) I: $s = 4$

(d) II: $s = 2$ (e) II: $s = 3$ (f) II: $s = 4$

(g) III: $s = 0.6$ (h) III $s = 1$ (i) III: $s = 4$

(j) IV: $s = 1$ (k) IV: $s = 2$ (l) IV: $s = 4$

Figure 7.15: Some examples of different curves. (a) (b) (c): curly curves; (d) (e) (f): irregular curves; (g) (h) (i): closed curly curves; (j) (k) (l): s-shaped curves.

returns 1 if $P(t)$ is true and -1 otherwise; and $P(t)$ is a function of time which keeps checking the amount of angle change of the curve between the initial angle and the current angle. If the angle change is greater than the current angle from the field, $P(t)$ is false; otherwise, $P(t)$ is true. $Random(0.0001, 0.01)$ returns a random floating point between 0.0001 and 0.01, which provides an unexpected value for the function $f(t)$. This s-shaped function $f(t)$ provides increasing values with random intervals and when the angle change of a particle attains a value (say 45 degrees), the $f(t)$ function has decreasing values randomly. During decreasing, if the angle change attains a value, the decreasing stops and the function $f(t)$ begins to increase again. In summary, this function provides a set of positive random values and then a set of negative random values. In this way, the curvature alternates from a negative value to a positive value, which forms an S shape. In Figure 7.5 (c) and (d), notice that it is difficult for us to detect any influence from the direction assignment (D1 and D2); instead, the small-scale properties of the s-shaped curves dominate the visual appearance.

7.4.1 Splitting Technique

We propose a splitting technique to produce cracks, a type of natural patterns. The cracking is similar in spirit to the work by Federl [38] and by Neff and Fiume [127], but not based on physics. Our process randomly places an irregular trail using irregular curve type (as shown in Figure 7.15 (d), (e), and (f)) in the region first. These irregular curves are shown in Figure 7.5 (a), (b), (e) and (f) as well. We suggest controlling the splitting under length control and stage control. For length control, we suggest if the arc length of a curve exceeds a length threshold, the system seeds new particles evenly at Δl separation (see Figure 7.16). New particles are emitted along the parent curve, alternating sides. Their directions maintain a certain angle to the tangent direction of the parent curve. Initially, each particle is allowed to travel for several timesteps and stops when it intersects with other curves or the region boundary. The trail is kept if the distance the particle travelled exceeds the length threshold Δl. For stage control, we suggest after we have the first curve, the process repeats for a known number of stages and stops no matter what the partition is. In each timestep, a curve emits a fixed number (n) of particles along the current curve

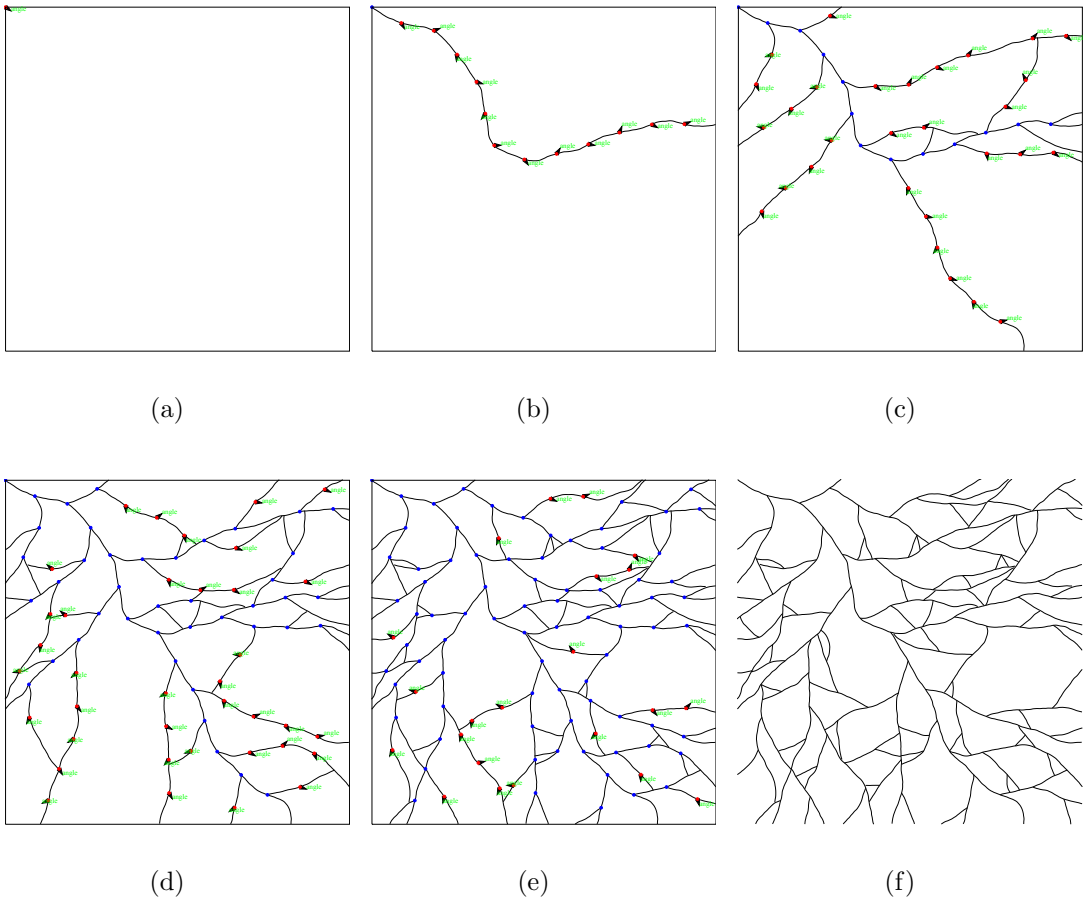
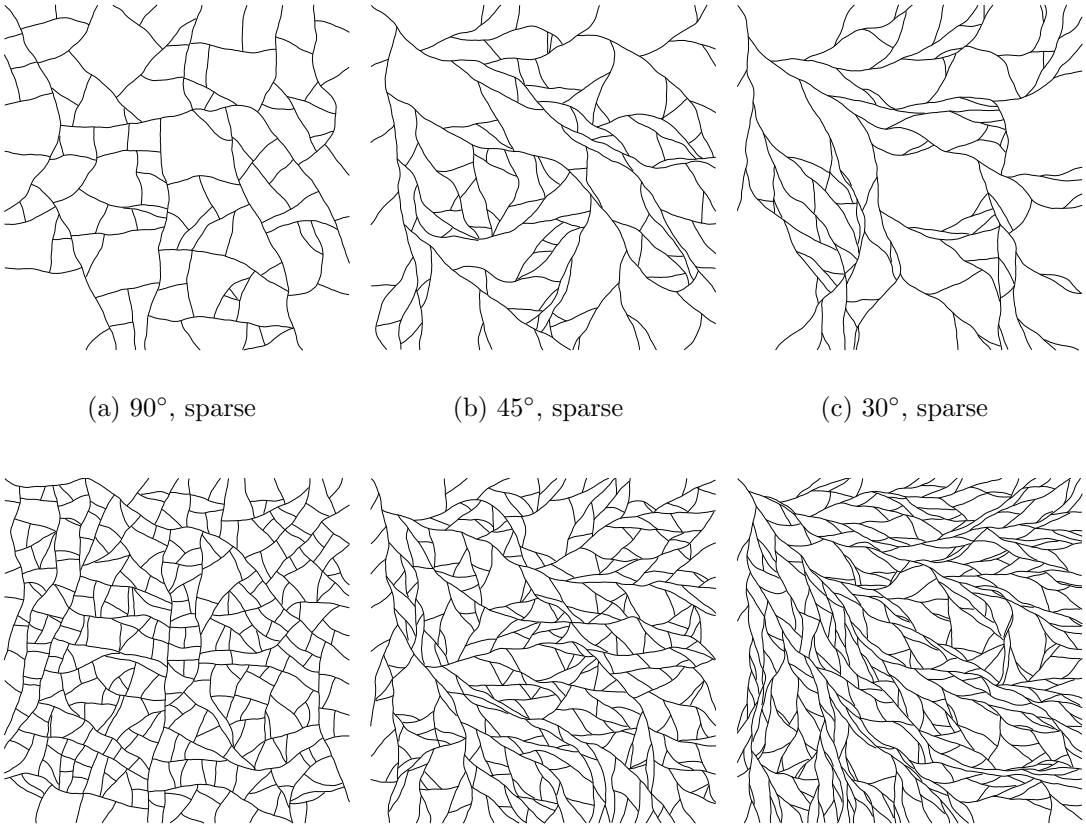


Figure 7.16: The process for splitting under length control. (a) A particle starts at the left top corner of the region. (b) It runs irregularly and randomly over the region and stops when it meets with the region boundary. The trail of the particle is the first curve placed. (c) If the curve length is greater than the threshold, along the curve, the system generates a few new particles. The new particles run randomly until new trails collide with other curves and the region boundary. (d) and (e) If the new curves are still greater than the threshold, the system generates a few new particles and grow new curves again. (f) The system stops because no curves are longer than the threshold. The tessellations complete.

and only stops when the number of stages reaches the maximum, specified by users.

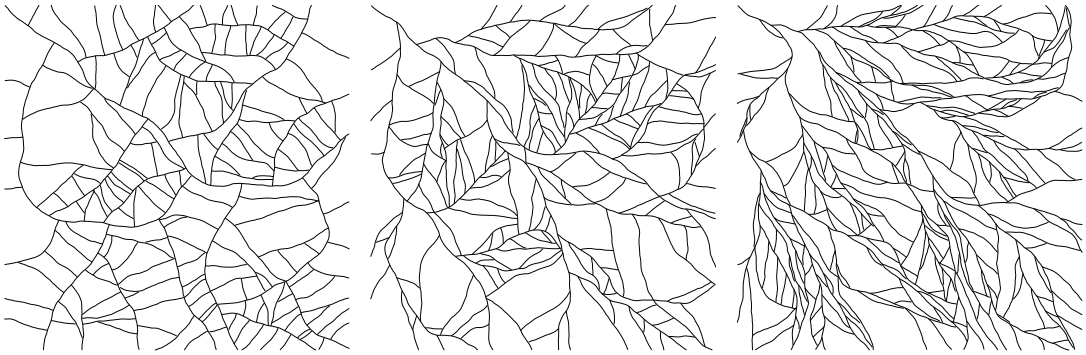
Figure 7.17 shows a few cracking examples by the splitting method under length control. Sparse versions using larger Δl are shown on the top, while dense tessellations employing smaller Δl are shown in the bottom row. Figure 7.18 provides splitting results under stage control (n = 7). This process introduces irregularities and randomness into the final results because of the use of irregular curves and the unexpected intersections. The randomness in the splitting process makes it difficult



(a) 90°, sparse (b) 45°, sparse (c) 30°, sparse

(d) 90°, dense (e) 45°, dense (f) 30°, dense

Figure 7.17: Splitting from length and angle control using irregular curves. Top: $\Delta l = 100$; bottom: $\Delta l = 50$.



(a) 90° (b) 45° (c) 30°

Figure 7.18: Splitting from stage control.

to control the spatial appearance. Even spatial distribution may be lost (e.g., Figure 7.17 (e)) and sometimes big empty areas appear in the partition. However, each child curve is grown with the same starting angles along its parent curve, and the same angle shows frequently all over the region. Humans have the abilities to easily detect the similarities and group all similar curves into a texture. The suggestion for length control provides spatial adjustment by removing the curves that can create narrow spacing. The results from length control look better than those from stage control. Stage control may be suitable for representing branching phenomena.

The splitting technique can be used to create shapes reminiscent of cracks or leaves. Federl [38] mentioned that cracks in dried mud commonly split at 90 degrees. Figure 7.17 (a), (d), and Figure 7.18 (a) split child curves along 90 degrees to their parent curves and resemble this cracking phenomenon. Figures 7.17 (f) and 7.18 (c) look more like leaves. Figure 7.1 (b) displays a simple result by overlaying the cracks on top of a painting generated by the image parsing method proposed by Zeng et al. [202]. This splitting result simulates the craquelure often seen in old oil paintings.

We compare our particle-based cracks with the results by a physical system that was proposed by Neff and Fiume [127] in Figure 7.19. The cracks in Figure 7.19 (b) are created by randomizing parameters on base patterns in Figure 7.19 (a). The base patterns are constructed with blast curves that depict the force-loading profile of an explosive material. Our cracks are generated by applying similarity in starting angles during curve growth. Figure 7.19 (c), (d), and (e) show our different cracks using length control. Our cracks grow from centers towards the region boundary and look more varied than Neff and Fiume's cracks. In addition, our method easily lends itself to many applications, not only for creating cracks, but also for the generation of other natural patterns, such as leaves and scales (mentioned later).

7.4.2 Stacking Technique

Scales

The tile shapes from the splitting technique are sometimes too random and irregular. We plan to use the tessellation method for generating more regular and organized patterns. We propose a technique called stacking, also based on the basic sequential idea. The stacking technique can create patterns resembling scales on snakes or fish.

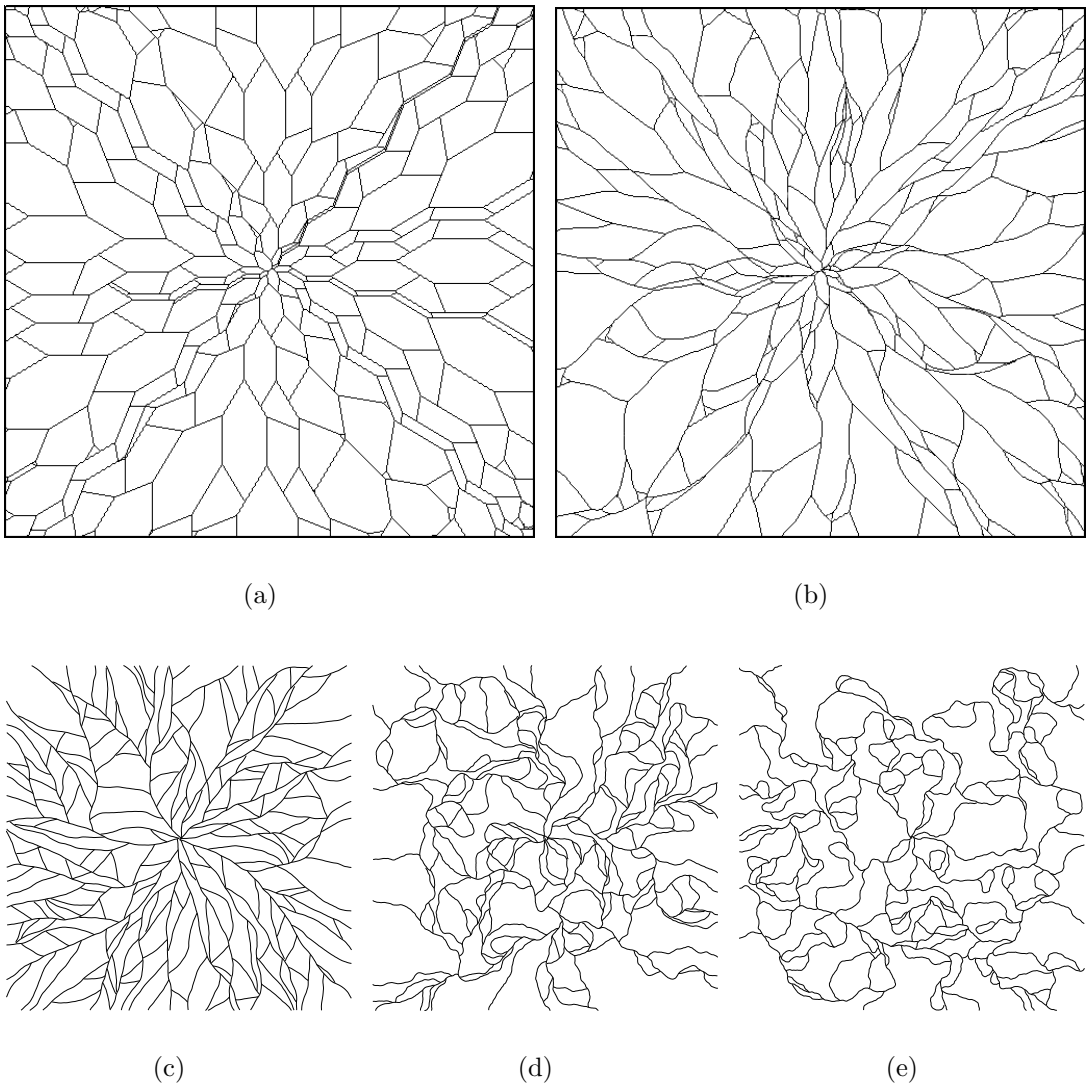
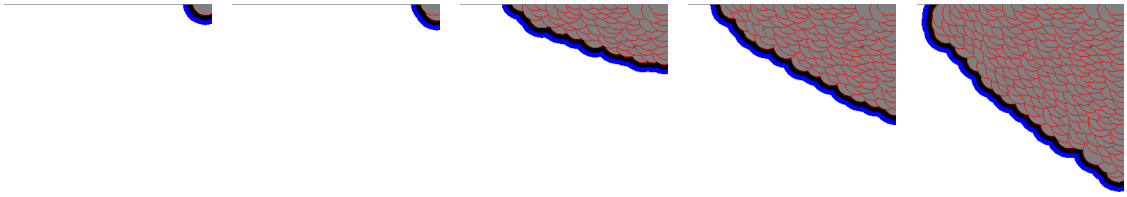


Figure 7.19: Comparison of our particle-based cracks with a physical system by Neff and Fiume [127]. (a) The base crack patterns by Neff and Fiume; (b) cracks from random variations by Neff and Fiume; (c) our cracks from length control; (d) and (e) our cracks using the same scheme in (c) but with different random variations.

Given a region and a growth path, we create a priority map for all pixels. The priority is the distance to the starting point. A pixel with a high priority is processed earlier than a pixel with a low priority. At each step, the system finds the location with the highest priority and grows a curve from that location. The curve is called a scale. We only keep curves with arc length in a proper range; otherwise, the system removes the curve and starts a new curve at the next location. We use an identification map that

stores the curve ids for each position. The identification map helps us quickly find the next candidate position. After a valid curve is found, the new scale is drawn. The points inside the scale and on the curve become invalid, shown in Figure 7.20 as grey and red regions respectively. The dark regions indicate the points that are too close to previous scales, where new curves are not allowed to begin. The points marked in blue have high priority. Each scale is grown step by step along the stacking order and direction. Figure 7.20 illustrates the generation process for the scales, following the curved path shown in Figure 7.21 (d) and starting from the right corner and growing from top to bottom. The growth of curves in Figure 7.21 (b) and (c) follows a vertical path from the top to the bottom with circular and wavy curves respectively. Figure 7.21 (e) and (f) show two stacking results with the same curved path but with different sizes of scales.



(a) 1 (b) 2 (c) 70 (d) 140 (e) 250

Figure 7.20: The growth for scales.

Rivers

Numerous elongated textures are also commonly seen in nature, such as rivers, hair or feathers. Previous region-based mosaic methods [50, 103, 104, 163] pack regular tiles into regions, which cannot show the flow of the texture. We propose to use the stacking technique for the creation of rivers. For both wavy and curly rivers, stacking starts with a random distribution of points. Each point grows a horizontal wavy or curly curve until the curve meets with previous curves or the region boundary. Stacking goes incrementally from bottom to top. If a particle is too close to the previous curves, it dies. We remove short curves. Each curve represents a small portion of the water surface. The top row of Figure 7.22 shows two illustrations of a wavy river. We create the bottom row in the same way as the top row. Instead of

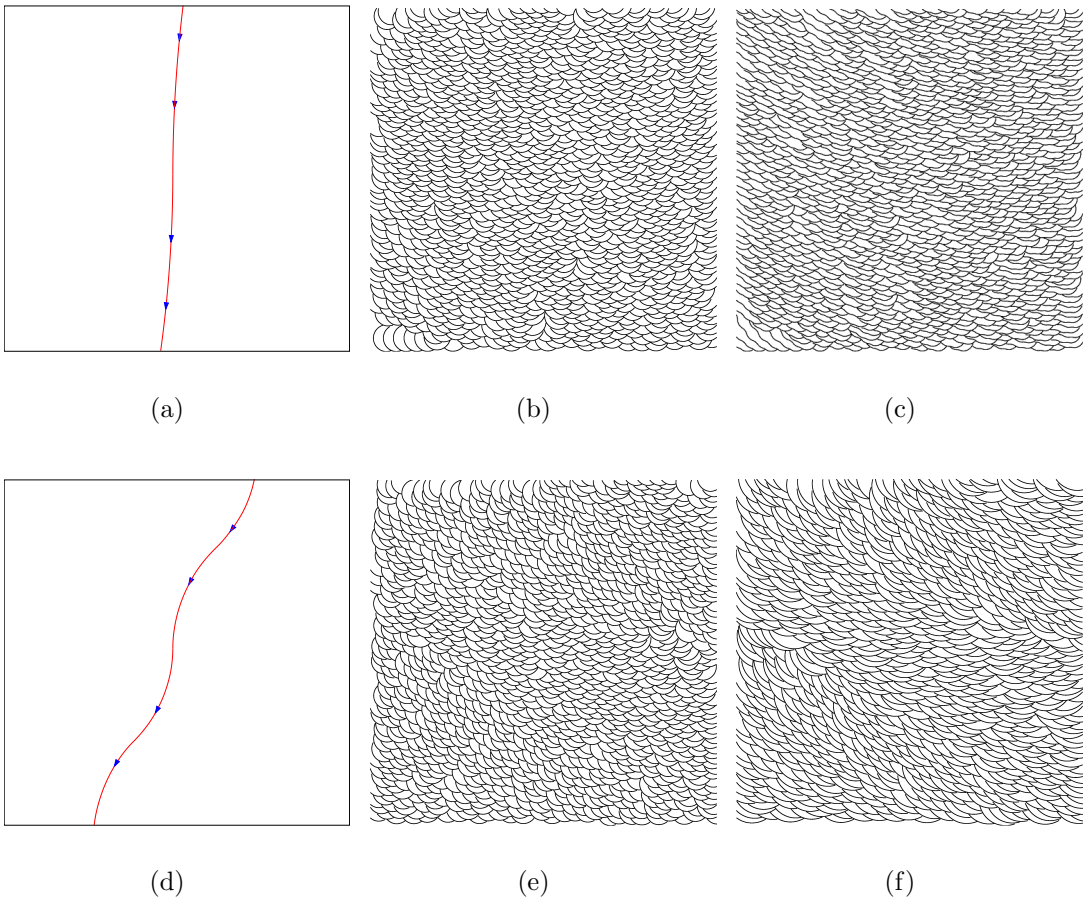


Figure 7.21: Four scale examples. (a) The stacking direction for the top row; (b) Circular-like scales; (c) Wavy scales; (d) The stacking direction for the bottom row; (e) Small scales; (f) Large scales.

placing one curve at each location, we place a group of curves. Each group includes three to six curves with similar properties, including similar starting directions and curvature, in which a random variation is added. The stacking goes group by group and the system stacks a group of curves from bottom to top. Some curly waves are shown at the bottom of Figure 7.22.

3D Indication

Next, we propose to indicate 3D illusions by growing curves in 2D. The idea behind the basic tessellation is to form regions with curves so that the regions form a smooth surface. A region in 2D can be thought of as a projected area from an object's surface in 3D. Stacking a group of curves can imply the shape of a surface in 3D. When a 3D

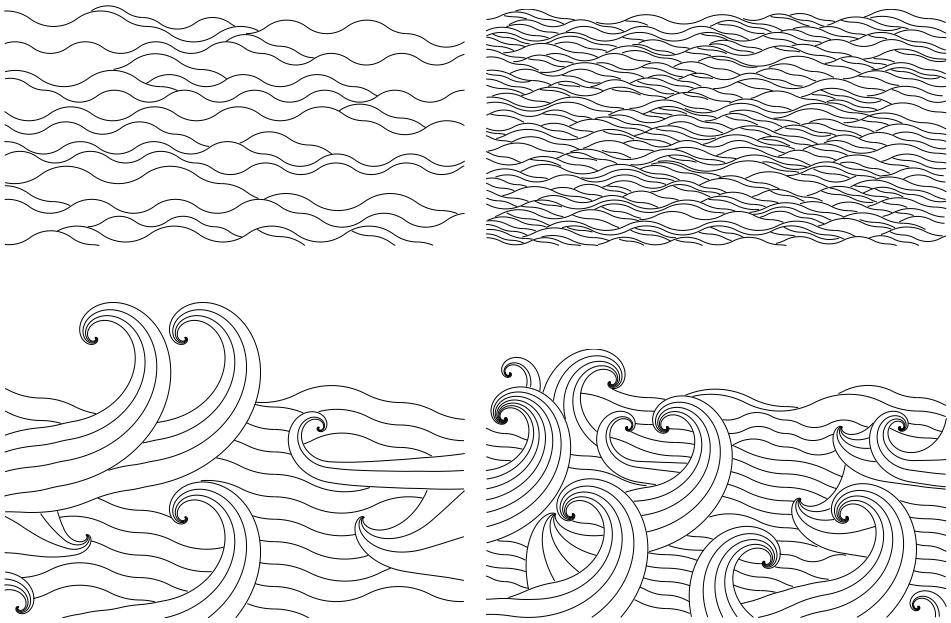
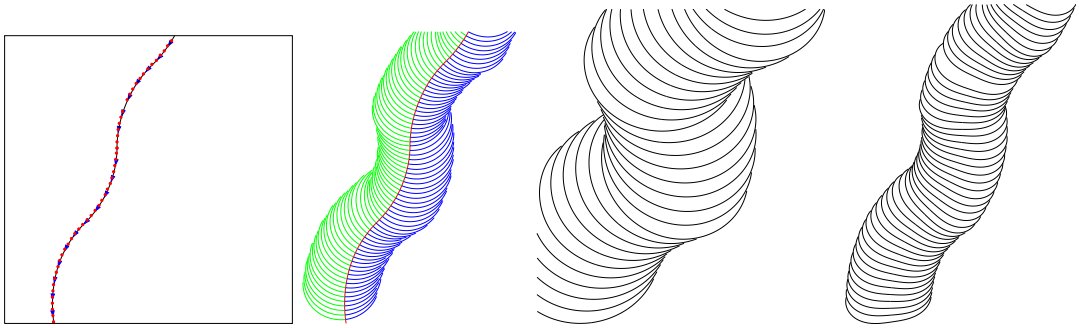


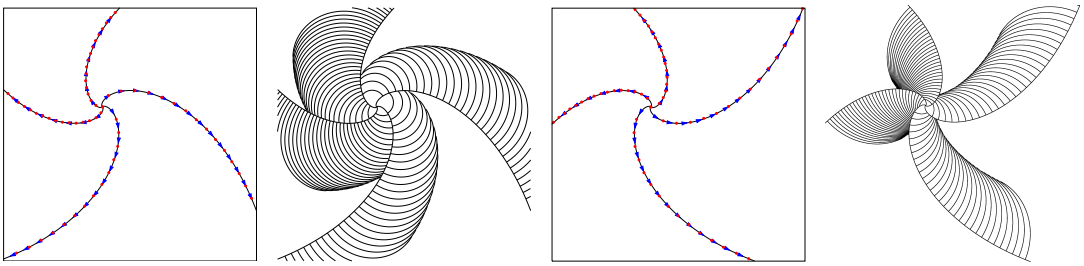
Figure 7.22: Sparse and dense textures for wavy and curly rivers.

surface is sliced with a plane along a chosen axis repeatedly, the collection of slices is a group a curves. If we project the collection of curves to 2D, the curves suggest the shape of surface. Specifically, we can define a stacking path and order and place curves along the stacking path one by one. Figure 7.23 (a) shows a stacking path (the red curve) and an order, shown by the arrows. A simple construction is to stack a group of circular curves along this configuration line sequentially, which generates the illusion of a 3D cylinder, shown in (b). Figure 7.23 (c) uses a large radius, displaying a group of larger circular curves that convey the impression of a larger cylinder. We also show an illusion of a small flat cylinder in (d), which is obtained by a slightly flatter curve. The flatter curve is generated by $f(t) = 0.001 * (t - ((int)(t/500)) * 500)$, called closed curly curves. At the beginning, the curve increases its curvature and decreases its curvature at the end. The constant value 500 used in the function $f(t)$ is to control when to reduce the curvature. Stacking contours can be complicated by a careful layout of curves; also, we can have varying curvature, e.g., as shown in Figure 7.24. The results convey 3D illusions for a sense of even more complicated surfaces. The concept of 3D suggestion is a way to obtain interesting visual appearance.

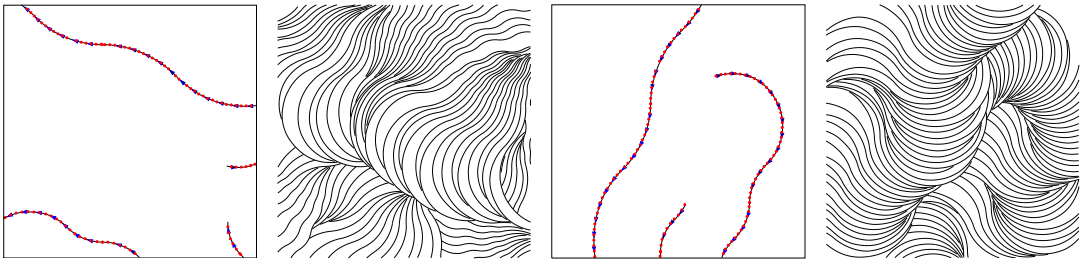


(a) (b) (c) (d)

Figure 7.23: 3D indication from stacking. (a) A stacking path; (b) circular curves (curly curve type, $s = 4$); (c) circular curves (curly curve type, $s = 1$); (d) flattened circular curves (closed curly type, $s = 4$).



(a) C1 (b) C1 (c) C2 (d) C2

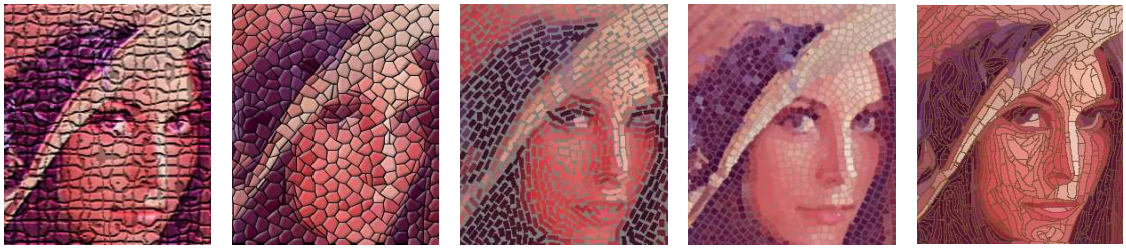


(e) C3 (f) C3 (g) C4 (h) C4

Figure 7.24: 3D indication. (a), (c), (e), and (g) are stacking configurations. (b) Curly curves (scale = 0.2); (d) the same curve as (b); (f) and (h) more complicated curves.

7.5 Results and Discussion

This chapter first explored artistic tessellation, applied it to produce mosaics with texture indication, and synthesized abstract and natural patterns automatically. Texture



(a) (b) (c) (d) (e)

Figure 7.25: Comparisons with previous methods. (a) Adobe Photoshop; (b) GIMP; (c) Hausner's method; (d) artificial mosaics; (e) our method ($A = 2$, smoothed twice).

indication in mosaics was not attempted by previous methods. We use a particle system and a smoothed vector field to create partitions, providing texture indication for mosaics. Figure 7.25 compares our result with commercial software (Adobe Photoshop and GIMP), and previous region-based methods such as traditional mosaics [50] and artificial mosaics [27]. Notice that no previous mosaic results can preserve highly textured areas, for example, Lena's hair or feathers. Our result in Figure 7.25 (e) depicts the hair texture nicely, showing hairs intersected with each other, despite using a smaller number of bigger tiles. The tiles are less attractive in smooth areas, such as Lena's face. The vector field in the face region lacks strong directions and the directions change quickly in that area. When particles trace that area, the trails will change quickly and smoothing cannot relieve the quick change. That is why we can find strange tiles on the face region. Human perception is sensitive to the face region because we are very familiar with the face and know exactly how it should look. Our current method still needs to improve the tile shapes in such regions by introducing an artificial strong direction to remove the strange tiles.

We show several applications including natural and abstract patterns, mosaics, and 3D suggestions, which demonstrate tremendous flexibility to achieve different purposes. Most applications we shown are based on the sequential method because the S-Method can determine each location of growth as well as when to stop creating new particles. The parallel method is based on an existing distribution and can produce regularity in the tessellation, which can be used for the mosaic goal as well. Figures 7.26, 7.11 (b), and (c) use the same distribution to place curves. The final

quality of P-Method highly depends on the initial distribution. In summary, the P-Method can provide a partition with better expression of the direction field and the S-Method is better able to produce long curved tiles.

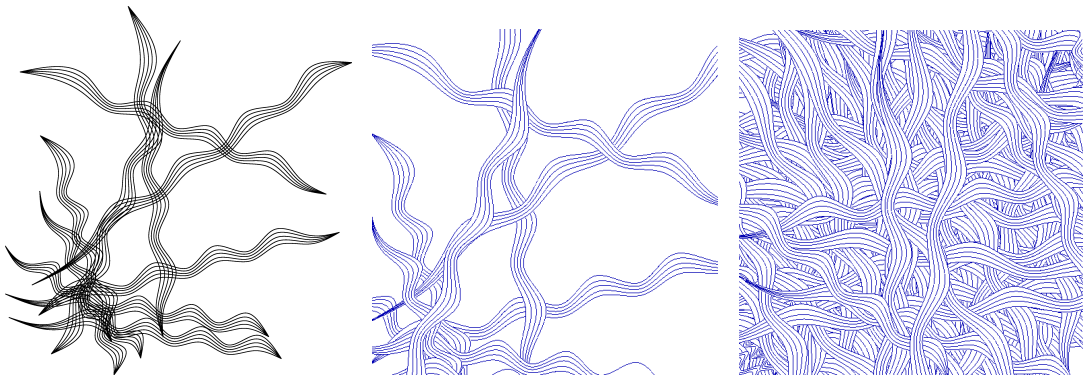
Figure 7.26: Mould's stippling with P-Method. Left: tessellations; right: tiles are assigned the average color of each region.

Beyond abstract and natural patterns, the idea of artistic tessellation may work for creating textures such as hair, feathered wings, flowers, and other botanical elements. In the future, users and researchers can adapt our method to those additional applications.

We use the stacking technique to create some examples of illusory three-dimensional structures that indicate 3D. Occlusion occurs when one object is hidden by another object partially or completely. From occlusion, we can infer how two objects are located. The object occluded is farther than the one unoccluded. The curve termination rules in our tessellation imitate occlusion. Many existing non-photorealistic rendering techniques start by finding silhouettes from 3D models or image edges and then draw inner content of objects. In contrast, the stacking technique starts with drawing interiors of an object by placing a group of curves and then forms the illusion of the surface of the object. The stacking technique constructs apparent objects with

a procedural organization of curves. We did not explore the management of intersections in this thesis much. We think that the intersection adjustment between curves is a very intriguing future direction.

In Figure 7.27 (a), some groups of wavy curves are drawn randomly in a region. Each group forms a long leaf. A simple intersection organization could be to always place a leaf on top of other leaves; a new leaf occludes other leaves, providing a striking depth illusion. Figure 7.27 (b) shows how near leaves obstruct far leaves, giving us a very strong feeling of depth. Figure 7.27 (c) employs many leaves occupying the entire region. Future exploration can provide a strategy to control the intersections when two curves collide each other.



(a) (b) (c)

Figure 7.27: 3D illusion from exclusion controls.

In our implementation, a discrete map is stored to help the intersection checkup. The strategy always lets each particle run a few initial steps (say 5). Otherwise, intersections with the parent curve will occur too frequently. The more initial steps are allowed, the more likely it is that there will be an intersection. An area with too many particles may disobey the principles of artistic tessellation since small and narrow regions will show. Figure 7.11 (c) has very dark areas on the bottom of the left corner, where curves intersect with other curves without stopping the growth. If the number of particles is very high, the particles in that area will create short curves. Since particles run at least five timesteps, particles may collide with each other during the initial steps. To improve the partition quality in very dark regions, we suggest lightening the whole image. Our current system controls the starting orientation of

Image	resolution	points	curves	tiles	time (s)
Figure 7.11 (b)	400×468	9,803	3,164	3,579	5.5
Figure 7.26	400×468	9,803	12,608	8,750	13.9
Figure 7.12 (d)	400×468	17,997	3,470	3,554	7.2
Figure 7.1 (d)	500×326	17,832	2,335	2,523	8.6
Figure 7.14 (d)	660×560	17,300	5,385	6,997	23.7
Figure 7.13 (b)	622×800	16,791	6,046	7,298	29.5
Figure 7.13 (a)	800×935	19,571	13,228	15,613	54.8

Table 7.1: CPU processing time for mosaics.

each particle, but does not predict the stop condition. How curves should meet with other curves is unpredictable. A new strategy for curve generation should consider where and how to terminate the growth. There is no need to make a change in the underlying method. Since we employ a discrete map to allow the system to terminate the growth of curves in a shorter time and thus avoid narrow spacing between two curves, the ending points do not really intersect with previous curves. We checked a 8-connected neighborhood. As a consequence, we may notice some vector graphics are not closed. We suggest in the future we may let the system grow without stopping at intersections and improve the final partitions in post processing by removing narrow regions.

Using an Intel Core Duo CPU E8400@ 3.0GHz with 3GB RAM, we can produce abstract patterns or cracking results in around $0.1 - 0.5$ seconds. If a distance map is used for spatial adjustment, processing takes longer, up to a minute. Table 7.1 shows the CPU processing time for mosaic generation.

7.6 Summary

This chapter presented the artistic tessellation technique to grow an irregular tessellation by placing curves in an organized way. The placement of curves presents similarity with spatial control. We implement the idea through a particle system. The key novelty is to determine a partition by placing curves. The generation of the partition is fast. In addition, the particle tracing strategy provides control over the final result. We have used this idea to produce natural and abstract patterns including cracks, leaves, waves, scales, and 3D illusions. We also introduced a texture-indication

mosaic method. This kind of tessellation should be very useful for irregular partitions.

This work on artistic tessellation has been published as follows:

- Hua Li and David Mould. Artistic tessellations by growing curves. In *Proceedings of the 6th international symposium on Non-photorealistic animation and rendering*, Non-photorealistic animation and rendering '11, pages 125–134, New York, NY, USA, 2011. ACM

Chapter 8

Conclusion

This thesis investigates a variety of styles, including halftoning, screening, stippling, line drawing, and artistic partition, by using two techniques: first, a contrast-aware and second, prioritized error diffusion algorithm and an artistic tessellation approach. The prioritized error diffusion method processes pixels in an order based on priorities and diffuses error to neighbouring pixels in a contrast- aware weight distribution. The use of priorities and the contrast-aware error diffusion helps preserve structural content from an image. The tessellation method produces artistic partitions by growing curves. We grow curves with similar starting direction and curvatures and prevent narrow spacing as well. We apply both algorithms to create several effects, including halftoning, screening, stippling, line drawing, and mosaics. The stylization applications used in this thesis demonstrate that consideration of contrast awareness and similarity in our NPR algorithms can provide high quality structure preservation.

8.1 Summary of Results

This thesis provides an automatic image abstraction system using Prioritized CED (introduced in Chapter 3) and its variations. This priority scheme can flexibly adapt to the framework of different primitives, including pixels, dots, and lines, all of which maintain contrast awareness in structure preservation. The applications for effects and styles included halftoning, screening, stippling, and line art. Our methods produced high quality stylized images from input images with a computation time comparable to earlier abstraction systems. The successful stylized effects for halftoning, screening, stippling, and line art demonstrated that the idea of contrast consideration can improve the abstraction quality. The variations of these different styles, shown in Chapters 4, 5, and 6, modify mask properties, organize the priority configuration, and adjust the weight distribution related to the element size. Each chapter evaluated our algorithms through side-by-side comparisons and through comparing

computation speed and objective measurements. Our evaluations demonstrated that our proposed automatic abstraction system can effectively accomplish the goal of high quality image abstraction with respect to tone matching and structure preservation.

The approach of Prioritized CED, introduced in Chapter 3, provides control over contrast by using a priority-based framework for error diffusion. We can produce stylized results efficiently. The application in Chapter 3 to halftoning generation results in high matching between rendered images and the original images in terms of tone, structure, contrast, and visual appearance. We provide a comprehensive comparison with previous halftoning methods using side by side visual views as well as objective measurements. The visual comparison shows that our contrast-aware halftoning can preserve weak edges, which are normally missed by previous algorithms. The testings from the objective measurements show that our methods have better scores as measured by CPSNR and MSSIM in general.

Chapter 4 adapts Prioritized CED to create screened results. Our screening method can preserve structural details well and provide multiple controls over patterns. We can create exclusion patterns or textural patterns according to two variations: modifying the circular masks to exclude some pixels in the error diffusion process and configuring multiple stages for different feature edges. Our screening results have better structural preservation than the dithering method [13, 87]. The flexible choices on pattern generation provide screenings with non-uniform appearance.

Chapter 5 proposes a stippling method with good structure matching to the original image, which adapts Prioritized CED by introducing additional control over the number of primitives used. The adapted approach for stippling can reduce the number of dots to a degree but still maintain the good preservation of content in image input. Additional applications based on variations on mask shapes, priority configuration, and colors can produce a family of styles, such as heightening and scratchboard. As compared with previous stippling methods [117, 161], our stippling method can preserve structure better. In addition, our method can produce a variety of effects.

Chapter 6 proposes an approach for creating line drawing. We adapted Prioritized CED by adding techniques of pixel clustering and skeletonization to produce a line drawing with simplified large-scale structure from an input image. Variants include

simplified, emphasized, or combined line art.

Chapter 7 presented an automatic system to create different artistic tessellations, which simulated natural patterns and varied artistic designs in 2D space. The system can be easily adapted to generate stained-glass mosaics when guided by an image. Different tessellations were constructed by growing curves procedurally and in parallel. The curves are the trails of particles in a particle system. The particle system presented similarity in the starting angle and movement types of the particles by locally controlling particle properties. The tessellations created by our curve growth can simulate natural and abstract patterns, including cracks, scales, rivers, and 3D illusions. Among these applications, cracks and scales contain irregularities and randomness, which effectively model natural patterns.

In the chapters about CAH, SPS, line drawing, CSS, and AT, we compared our stylized results with previous results created by other researchers. We used the same original images as they did in their papers in order to have fair side-by-side comparisons with their results. This thesis shows a small number of our results. Beyond this thesis, we have produced around 200 rendered images for each of our proposed methods. Test images included different subjects such as cats, cars, buildings, flowers, persons, mugs, and birds. Based on the resulting images, our CAH, SPS, line drawing, and AT methods are very robust. The images created by CSS look good too, but CSS results have assignment errors because we cannot recognize two different regions belonging to the same object from segmentation. This is an unavoidable issue which happens to other researchers too.

Throughout this thesis, we suggested and justified the parameter values. Our testings showed that the parameter values used are not sensitive to the image contents.

8.2 Future Work

Even though the work presented in this thesis is quite comprehensive, the research undoubtedly reveals numerous possibilities for extending the work that has been done. The remainder of this section describes some new potential research directions.

Contrast-aware error diffusion may be applied to another style such as black and white stylization, which is an extreme contrast. Black and white style separates foreground from background. Prioritized CED offers a step to threshold intensities,

which may produce numerous isolated pixels, much like the halftoning. We should find a way to cluster black pixels and to resolve the issue concerning scattered pixels. Similar to the black and white style, black and white engraving is another possible style that would extend this work through strokes. As for the priority scheme, we think that is a general idea which will have other applications in graphics beyond NPR.

Introducing semantic information into our artistic tessellation method can be useful for creating partitions with meanings such as landscape or portrait. We may think about how to accurately describe semantic content when growing curves. Another possible extension could be the investigation on occlusion using curves that may create interesting 3D illusions in the future.

In Chapter 3, the contrast-aware halftoning method processed faster than previous structure-aware halftoning methods. However, our contrast-aware halftoning method might produce a group of black pixels close together. In the future, an additional post process can help to remove clumping and improve the visual appearance. The influence of the shape of the mask on the effects should be researched further and will likely produce interesting findings. Other image features such as local histogram, texture features, entropy, and statistical factors may be very useful to adjust the local contrast. We created color halftoning by simply extending one color channel to the RGB color channels. Other possible areas of expansion that emerge in Chapter 3 would be the introduction of color halftoning and other artistic styles generated through pixel management provide further research topics. The speed can be further improved by a parallel system.

In Chapter 4, the content-sensitive screening method worked well for a variety of images; however, CSS did not address the matching issue and thus incorrect assignments happened. In the future, the assignments of patterns to regions should be explored further as not every pattern offers an aesthetic look.

In Chapter 5, the structure-preserving stippling method calculated the priority values based on the distance to two extreme intensities. In the future, the exploration of other types of priorities for the SPS method would be interesting. For example, we can prioritize based on values from local histogram, steepness, or slope of edges. Mask variations on shapes and growth order provide many possibilities of a variety

of pattern generations.

In Chapter 6, we explored two kinds of line art by extending structure-preserving stippling. We did not consider the traditional hand-drawn line art from media such as pencil, pen, and charcoal. In the future, we can apply example-based methods by using reference works from real artists to improve the visual appearance.

In Chapter 7, artistic tessellation uses particle tracing to create curves. Particle tracing has been frequently used in animation applications. In the future, in particular, we would like to extend the drawing approaches to animating line art. Also, studying groups of particles may be very interesting. Managing occlusions may offer very convincing illusions of 3D shapes.

8.2.1 Example-Based Methods

Our Prioritized CED is effective for preserving structure in image abstraction; however, there are many traditional styles that have unique visual appearances resembling a hand-drawn look. When stippling, artists may create dots or small strokes that vary in shape and size. Our SPS generates perfect circles, which is distinct from the aesthetics of hand-drawn work. The drawings that we presented ignored the visual look of strokes. However, the real drawings use a medium, such as pencil. Each stroke of the pencil is an interaction between the pencil and the canvas. The force applied on each stroke can affect the appearance too.

Example-based methods [4, 58, 59, 66, 86, 117] have been effectively applied to simulate the realistic look of media in NPR. Kim et al. [86] and Martín et al. [117] also adapted example-based methods to create stipplings with a hand-drawn appearance. A future improvement would employ example-based methods in our algorithms. For hand-drawn stippling generation, we can use Prioritized CED to obtain the dot distribution. Based on the distribution, we can place a stylized dot resembling the hand-drawn look after analyzing the statistics of artists' works. Since the structure is well preserved, the realistic appearance of the styles will certainly enhance the quality of stylization in terms of artistic aesthetics. However, when the examples are applied, they may affect the structural quality.

8.2.2 Interactive Tools

Another useful extension of this work would be to transform our automatic systems into interactive tools. Computers have become a popular means of creating art; our tools could provide artists with an interactive system for convenient editing and management.

We believe that our automatic systems can be directly transformed into interactive tools for users by providing interactive operations for selecting important objects and supporting convenient parameter configurations manually. The ability to select the most important structure for stylization can further enhance the structural preservation. Especially, the system for tessellation can provide different procedural configuration for users. Users can specify specific growth of curves for each region and different objects can have different tessellation configurations, both of which would result in interesting patterns.

8.2.3 Long-Term Future Work

In the long term, future work should consider the relationship between art, perception, and algorithms. Color perception is an obvious possible addition to improve stylization quality. The contrast between colors is complex. How can we maintain harmony in colors and in color contrast? Nevertheless, this new direction is an interesting trend in NPR research. This thesis explored a few perceptual rules including contrast and similarity. There are many principles of perceptual organization, such as grouping and simplification, which are very useful for algorithm designs and provide control over placement of primitives. Appendix A offers a brief introduction to the principle of grouping.

Bibliography

[1] Mostafa Analoui and Jan P. Allebach. Model-based halftoning using direct binary search. In Bernice E. Rogowitz, editor, *Proceedings of SPIE*, volume 1666 of *Human Vision, Visual Processing, and Digital Display III*, pages 96–108, 1992.

[2] Dustin Anderson and Zoë Wood. User driven two-dimensional computer-generated ornamentation. In *Proceedings of the 4th International Symposium on Advances in Visual Computing*, International Symposium on Visual Computing (ISVC '08), pages 604–613, Berlin, Heidelberg, 2008. Springer-Verlag.

[3] Farhan A. Baqai and Jan P. Allebach. Halftoning via direct binary search using analytical and stochastic printer models. *Image Processing, IEEE Transactions on*, 12(1):1–15, 2003.

[4] Pascal Barla, Simon Breslav, Lee Markosian, and Joëlle Thollot. Interactive Hatching and Stippling by Example. Research Report RR-6461, INRIA, 2006.

[5] Maurice S. Bartlett. *An Introduction to Stochastic Processes: with Special Reference to Methods and Applications*. Cambridge University Press; 3 edition, 1981.

[6] Sebastiano Battiato, Gianpiero Di Blasi, Giovanni Maria Farinella, and Giovanni Gallo. A novel technique for opus vermiculatum mosaic rendering. In *14th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision (WSCG'06)*, pages 133–140, Plzen, Czech Republic, 2006. UNION Agency-Science Press.

[7] William Baxter, Jeremy Wendt, and Ming C. Lin. IMPaSTo: a realistic, interactive model for paint. In *Proceedings of the 3rd international symposium on Non-photorealistic animation and rendering*, Non-photorealistic animation and rendering '04, pages 45–148, New York, NY, USA, 2004. ACM.

[8] Bryce E. Bayer. An optimum method for two-level rendition of continuous-tone pictures. *IEEE International Conference on Communications*, 1:26–11–26–15, 1973.

[9] Teresa W. Bleser, John L. Sibert, and J. Patrick McGee. Charcoal sketching: returning control to the artist. *ACM Transactions on Graphics*, 7(1):76–81, January 1988.

[10] Adrien Bousseau, Matt Kaplan, Joëlle Thollot, and François X. Sillion. Interactive watercolor rendering with temporal coherence and abstraction. In *Proceedings of the 4th international symposium on Non-photorealistic animation*

and rendering, Non-photorealistic animation and rendering '06, pages 141–149, New York, NY, USA, 2006. ACM.

[11] Adrien Bousseau, Fabrice Neyret, Joëlle Thollot, and David Salesin. Video watercolorization using bidirectional texture advection. *ACM Transactions on Graphics*, 26(3):Article No. 104, July 2007.

[12] Stephen Brooks. Image-based stained glass. *IEEE Transactions on Visualization and Computer Graphics*, 12(6):1547–1558, November 2006.

[13] John W. Buchanan. Special effects with half-toning. *Computer Graphics Forum*, 15(3):97–108, 1996.

[14] Sheelagh Carpendale. Information visualization. chapter Evaluating Information Visualizations, pages 19–45. Springer-Verlag, Berlin, Heidelberg, 2008.

[15] Jianghao Chang, Benoît Alain, and Victor Ostromoukhov. Structure-aware error diffusion. *ACM Transactions on Graphics*, 28(5):162:1–162:8, December 2009.

[16] Nelson S.-H. Chu and Chiew-Lan Tai. Moxi: real-time ink dispersion in absorbent paper. *ACM Transactions on Graphics*, 24(3):504–511, July 2005.

[17] Jonathan M. Cohen, John F. Hughes, and Robert C. Zeleznik. Harold: A world made of drawings. In *Proceedings of the First International Symposium on Non-Photorealistic Animation and Rendering*, pages 83–90. ACM Press, 2000.

[18] Daniel Cohen-Or, Olga Sorkine, Ran Gal, Tommer Leyvand, and Ying-Qing Xu. Color harmonization. *ACM Transactions on Graphics*, 25(3):624–630, July 2006.

[19] Dorin Comaniciu and Peter Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analalysis and Machine Intelligence*, 24(5):603–619, May 2002.

[20] Lin Cong, Ruofeng Tong, and Jinxiang Dong. Selective image abstraction. *The Visual Computer*, 27(3):187–198, March 2011.

[21] Cassidy J. Curtis, Sean E. Anderson, Joshua E. Seims, Kurt W. Fleischer, and David H. Salesin. Computer-generated watercolor. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '97, pages 421–430, New York, NY, USA, 1997. ACM Press/Addison-Wesley Publishing Co.

[22] Doug DeCarlo, Adam Finkelstein, Szymon Rusinkiewicz, and Anthony Santella. Suggestive contours for conveying shape. *ACM Transactions on Graphics*, 22(3):848–855, July 2003.

[23] Doug DeCarlo and Szymon Rusinkiewicz. Highlight lines for conveying shape. In *Proceedings of the 5th international symposium on Non-photorealistic animation and rendering*, Non-photorealistic animation and rendering '07, pages 63–70, New York, NY, USA, 2007. ACM.

[24] Oliver Deussen, Stefan Hiller, Cornelius van Overveld, and Thomas Strothotte. Floating points: A method for computing stipple drawings. *Computer Graphics Forum*, 19(3):41–50, 2000.

[25] Oliver Deussen and Tobias Isenberg. Halftoning and stippling. In Paul Rosin and John Collomosse, editors, *Image and Video-Based Artistic Stylisation*, volume 42 of *Computational Imaging and Vision*, chapter 3, pages 45–61. Springer, London, Heidelberg, 2013.

[26] Oliver Deussen and Thomas Strothotte. Computer-generated pen-and-ink illustration of trees. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '00, pages 13–18, New York, NY, USA, 2000. ACM Press/Addison-Wesley Publishing Co.

[27] Gianpiero Di Blasi and Giovanni Gallo. Artificial mosaics. *The Visual Computer*, 21(6):373–383, 2005.

[28] Mohamed Ould Djibril and Rachid Oulad Haj Thami. Islamic geometrical patterns indexing and classification using discrete symmetry groups. *Journal of Computer Cultural Heritage*, 1(2):10:1–10:14, November 2008.

[29] Murray Eden. A two-dimensional growth process. In *Proceedings of Fourth Berkeley Symposium on Mathematics, Statistics, and Probability*, volume 4, pages 223–239, Berkeley, CA, 1961. University of California Press.

[30] E. Bruce Goldstein (Editor), Glyn Humphreys (Consulting Editor), Margaret Shiffrar (Consulting Editor), and William Yost(Consulting Editor). *Blackwell Handbook of Perception (Blackwell Handbooks of Experimental Psychology) [Hardcover]*. Wiley-Blackwell; 1 edition, 2001.

[31] Kenneth R. Boff (Editor), Lloyd Kaufman (Editor), and James P. Thomas (Editor). *Handbook of Perception and Human Performance, Vol. 1: Sensory Processes and Perception*. Wiley-Interscience, First Edition edition, 1986.

[32] Otto G. Ocvirk (Editor), Robert E. Stinson (Editor), Philip R. Wigg (Editor), Robert O. Bone (Editor), and David L. Cayton (Editor). *Art Fundamentals: Theory and Practice [Paperback]*. Mcgraw-Hill (Tx), 9th Bk&Cdr edition, 2001.

[33] Gershon Elber and Elaine Cohen. Hidden curve removal for free form surfaces. In *Proceedings of the 17th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '90, pages 95–104, New York, NY, USA, 1990. ACM.

[34] Gershon Elber and George Wolberg. Rendering traditional mosaics. *The Visual Computer*, 19(1):67–78, 2003.

[35] Reiner Eschbach and Keith T. Knox. Error-diffusion algorithm with edge enhancement. *Journal of the Optical Society of America A: Optics, Image Science, and Vision*, 8(12):1844–1850, December 1991.

[36] Ricardo Fabbri, Leandro F. Estrozi, and Luciano F. Costa. On Voronoi diagrams and medial axes. *Journal of Mathematical Imaging and Vision*, 17(1):27–40, 2002.

[37] Geisa Martins Faustino and Luiz Henrique de Figueiredo. Simple adaptive mosaic effects. In *Computer Graphics and Image Processing, 2005. SIBGRAPI 2005. 18th Brazilian Symposium on*, pages 315–322, Washington, DC, USA, 2005. IEEE Computer Society.

[38] Pavol Federl. *Modeling fracture formation on growing surfaces.* PhD thesis, University of Calgary, Calgary, Alta., Canada, Canada, 2003.

[39] Robert W. Floyd and Louis Steinberg. An adaptive algorithm for spatial grey scale. *Proceedings of the Society for Information Display*, 17(2):75–77, 1976.

[40] Ronald H. Forgus. *Perception.* McGraw Hill; First Edition, 7th Printing edition, 1966.

[41] Weidong Geng. *The Algorithms and Principles of Non-photorealistic Graphics: Artistic Rendering and Cartoon Animation (Advanced Topics in Science and Technology in China) [Hardcover].* Springer; 1st Edition. edition, 2008.

[42] Andrew S. Glassner. Aperiodic tiling. *IEEE Computer Graphics and Applications*, 18(3):83–90, 1998.

[43] Andrew S. Glassner. Penrose tiling. *IEEE Computer Graphics and Applications*, 18(4):78–86, 1998.

[44] Andrew S. Glassner. Celtic knots, part 3. *IEEE Computer Graphics and Applications*, 20(1):70–75, 2000.

[45] Amy A. Gooch, Sven C. Olsen, Jack Tumblin, and Bruce Gooch. Color2Gray: salience-preserving color removal. *ACM Transactions on Graphics*, 24(3):634–639, July 2005.

[46] Bruce Gooch and Amy Gooch. *Non-Photorealistic Rendering.* A K Peters/CRC Press, 2001.

[47] Paul Haeberli. Paint by numbers: abstract image representations. *SIGGRAPH '90*, 24(4):207–214, September 1990.

[48] Nick Halper, Tobias Isenberg, Felix Ritter, Bert Freudenberg, Oscar Meruvia, Stefan Schlechtweg, and Thomas Strothotte. OPENNPAR: A system for developing, programming, and designing non-photorealistic animation and rendering. In *Proceedings of the 11th Pacific Conference on Computer Graphics and Applications*, Pacific Graphics (PG '03), pages 424–428, Washington, DC, USA, 2003. IEEE Computer Society.

[49] Lee Hammond. *Lifelike Drawing with Lee Hammond [Paperback]*. North Light Books, 2005.

[50] Alejo Hausner. Simulating decorative mosaics. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '01, pages 573–580, New York, NY, USA, 2001. ACM.

[51] James Hays and Irfan Essa. Image and video based painterly animation. In *Proceedings of the 3rd international symposium on Non-photorealistic animation and rendering*, Non-photorealistic animation and rendering '04, pages 113–120, New York, NY, USA, 2004. ACM.

[52] Christopher G. Healey. Combining Perception and Impressionist Techniques for Nonphotorealistic Visualization of Multidimensional Data. In Christopher G. Healey, editor, *Non-Photorelistic Rendering in Scientific Visualization*, SIGGRAPH 2001 Course Notes. ACM Press, New York, 2001.

[53] Christopher G. Healey, Laura Tateosian, James T. Enns, and Mark Remple. Perceptually based brush strokes for nonphotorealistic visualization. *ACM Transactions on Graphics*, 23(1):64–96, January 2004.

[54] Robert T. Held, Emily A. Cooper, James F. O'Brien, and Martin S. Banks. Using blur to affect perceived distance and size. *ACM Transactions on Graphics*, 29(2):19:1–19:16, April 2010.

[55] Aaron Hertzmann. Painterly rendering with curved brush strokes of multiple sizes. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '98, pages 453–460, New York, NY, USA, 1998. ACM.

[56] Aaron Hertzmann. Tutorial: A survey of stroke-based rendering. *IEEE Computer Graphics and Applications*, 23(4):70–81, July 2003.

[57] Aaron Hertzmann. Non-photorealistic rendering and the science of art. In *Proceedings of the 8th International Symposium on Non-Photorealistic Animation and Rendering*, Non-Photorealistic Animation and Rendering '10, pages 147–157, New York, NY, USA, 2010. ACM.

[58] Aaron Hertzmann, Charles E. Jacobs, Nuria Oliver, Brian Curless, and David H. Salesin. Image analogies. In *Proceedings of the 28th annual conference on*

Computer graphics and interactive techniques, SIGGRAPH '01, pages 327–340, New York, NY, USA, 2001. ACM.

[59] Aaron Hertzmann, Nuria Oliver, Brian Curless, and Steven M. Seitz. Curve analogies. In *Proceedings of the 13th Eurographics workshop on Rendering*, Eurographics Workshop on Rendering (EGRW '02), pages 233–246, Aire-la-Ville, Switzerland, Switzerland, 2002. Eurographics Association.

[60] Aaron Hertzmann and Denis Zorin. Illustrating smooth surfaces. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '00, pages 517–526, New York, NY, USA, 2000. ACM Press/Addison-Wesley Publishing Co.

[61] Stefan Hiller, Heino Hellwig, and Oliver Deussen. Beyond stippling – methods for distributing objects on the plane. *Computer Graphics Forum*, 22(3):515–522, 2003.

[62] Julian E. Hochberg. *Perception (Foundations of Modern Psychology Series)*. Prentice-Hall Inc.; 1st Edition / 8th Printing edition, 1964.

[63] Takeo Igarashi, Satoshi Matsuoka, and Hidehiko Tanaka. Teddy: a sketching interface for 3D freeform design. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '99, pages 409–416, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co.

[64] Tobias Isenberg. Evaluating and validating non-photorealistic and illustrative rendering. In Paul Rosin and John Collomosse, editors, *Image and Video-Based Artistic Stylisation*, volume 42 of *Computational Imaging and Vision*, chapter 15, pages 311–331. Springer, London, Heidelberg, 2013.

[65] Seok Jang and Hyun-Ki Hong. Stippling technique based on color analysis. In *Proceedings of the 6th Pacific-Rim conference on Advances in Multimedia Information Processing - Volume Part II*, Advances in Multimedia Information Processing (PCM'05), pages 782–793, 2005.

[66] Pierre-Marc Jodoin, Emric Epstein, Martin Granger-Piché, and Victor Ostromoukhov. Hatching by example: a statistical approach. In *Proceedings of the 2nd international symposium on Non-photorealistic animation and rendering*, Non-photorealistic animation and rendering '02, pages 29–36, New York, NY, USA, 2002. ACM.

[67] Randall W. Hill Jr. Perceptual attention in virtual humans: Towards realistic and believable gaze behaviors. In *Proceedings of the AAAI Fall Symposium on Simulating Human Agents*, pages 46–52, Menlo Park, Calif., 2000. AAAI Press.

[68] Robert D. Kalnins, Philip L. Davidson, and David M. Bourguignon. JOT. Software, 2002. `http://jot.cs.princeton.edu/`.

[69] Robert D. Kalnins, Philip L. Davidson, Lee Markosian, and Adam Finkelstein. Coherent stylized silhouettes. *ACM Transactions on Graphics*, 22(3):856–861, July 2003.

[70] Robert D. Kalnins, Lee Markosian, Barbara J. Meier, Michael A. Kowalski, Joseph C. Lee, Philip L. Davidson, Matthew Webb, John F. Hughes, and Adam Finkelstein. WYSIWYG NPR: drawing strokes directly on 3D models. *ACM Transactions on Graphics*, 21(3):755–762, July 2002.

[71] Henry Kang, Seungyong Lee, and Charles K. Chui. Coherent line drawing. In *Proceedings of the 5th international symposium on Non-photorealistic animation and rendering*, Non-photorealistic animation and rendering '07, pages 43–50, New York, NY, USA, 2007. ACM.

[72] Henry R. Kang. *Digital Color Halftoning*. Society of Photo-Optical Instrumentation Engineers (SPIE), Bellingham, WA, USA, 1999.

[73] Craig Kaplan. *Introductory Tiling Theory for Computer Graphics*. Morgan and Claypool Publishers; 1 edition, 2009.

[74] Craig S. Kaplan. Voronoi diagrams and ornamental design. In *The First Annual Symposium of the International Society for the Arts, Mathematics, and Architecture*, pages 277–283, San Sebastiá, Spain, 1999.

[75] Craig S. Kaplan. Computer generated Islamic star patterns. In Reza Sarhangi, editor, *Bridges: Mathematical Connections in Art, Music, and Science*, pages 105–112, Unknown, 2000. Tarquin Publications. Available online at `http://archive.bridgesmathart.org/2000/bridges2000-105.pdf`.

[76] Craig S. Kaplan. Islamic star patterns from polygons in contact. In *Proceedings of Graphics Interface 2005*, Graphics Interface (GI '05), pages 177–185, School of Computer Science, University of Waterloo, Waterloo, Ontario, Canada, 2005. Canadian Human-Computer Communications Society.

[77] Craig S. Kaplan. Metamorphosis in Escher's art. In Reza Sarhangi and Carlo H. Séquin, editors, *Bridges Leeuwarden: Mathematics, Music, Art, Architecture, Culture*, pages 39–46, Winfield, Kansas, 2008. Southwestern College. Available online at `http://archive.bridgesmathart.org/2008/bridges2008-39.pdf`.

[78] Craig S. Kaplan and George W. Hart. Symmetrohedra: Polyhedra from symmetric placement of regular polygons. In Reza Sarhangi and Slavik Jablan, editors, *Bridges: Mathematical Connections in Art, Music, and Science*, pages 21–28, 2001. Available online at `http://archive.bridgesmathart.org/2001/bridges2001-21.pdf`.

[79] Craig S. Kaplan and David H. Salesin. Escherization. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '00, pages 499–510, New York, NY, USA, 2000. ACM Press/Addison-Wesley Publishing Co.

[80] Craig S. Kaplan and David H. Salesin. Dihedral Escherization. In *Proceedings of Graphics Interface 2004*, Graphics Interface (GI '04), pages 255–262, School of Computer Science, University of Waterloo, Waterloo, Ontario, Canada, 2004. Canadian Human-Computer Communications Society.

[81] Craig S. Kaplan and David H. Salesin. Islamic star patterns in absolute geometry. *ACM Transactions on Graphics*, 23(2):97–119, April 2004.

[82] Matthew Kaplan and Elaine Cohen. Computer generated Celtic design. In *Proceedings of the 14th Eurographics workshop on Rendering*, Eurographics Workshop on Rendering (EGRW '03), pages 9–19, Aire-la-Ville, Switzerland, Switzerland, 2003. Eurographics Association.

[83] Matthew Kaplan, Emil Praun, and Elaine Cohen. Pattern oriented remeshing for Celtic decoration. In *Pacific Conference on Computer Graphics and Applications*, pages 199–206. IEEE Computer Society, 2004.

[84] Dongyeon Kim, Minjung Son, Yunjin Lee, Henry Kang, and Seungyong Lee. Feature-guided image stippling. *Computer Graphics Forum*, 27(4):1209–1216, 2008.

[85] Junhwan Kim and Fabio Pellacini. Jigsaw image mosaics. *ACM Transactions on Graphics*, 21(3):657–664, July 2002.

[86] Sung Ye Kim, Ross Maciejewski, Tobias Isenberg, William M. Andrews, Wei Chen, Mario Costa Sousa, and David S. Ebert. Stippling by example. In *Proceedings of the 7th International Symposium on Non-Photorealistic Animation and Rendering*, Non-Photorealistic Animation and Rendering '09, pages 41–50, New York, NY, USA, 2009. ACM.

[87] Donald E. Knuth. Digital halftones by dot diffusion. *ACM Transactions on Graphics*, 6(4):245–273, 1987.

[88] Johannes Kopf, Daniel Cohen-Or, Oliver Deussen, and Dani Lischinski. Recursive wang tiles for real-time blue noise. *ACM Transactions on Graphics*, 25(3):509–518, July 2006.

[89] Paul Kubelka. New contributions to the optics of intensely light-scattering materials. part i. *Journal of the Optical Society of America*, 38(5):448–448, May 1948.

[90] Paul Kubelka. New contributions to the optics of intensely light-scattering materials. part ii: Nonhomogeneous layers. *Journal of the Optical Society of America*, 44(4):330–334, Apr 1954.

[91] Nae-Joung Kwak, Soung-Pil Ryu, and Jae-Hyeong Ahn. Edge-enhanced error diffusion halftoning using human visual properties. *International Conference on Hybrid Information Technology, IEEE computer society, Washington*, 1:499–504, 2006.

[92] Jan Eric Kyprianidis and Henry Kang. Image and video abstraction by coherence-enhancing filtering. *Computer Graphics Forum*, 30(2):593–602, 2011.

[93] Adam Lake, Carl Marshall, Mark Harris, and Marc Blackstein. Stylized rendering techniques for scalable real-time 3D animation. In *Proceedings of the 1st international symposium on Non-photorealistic animation and rendering*, Non-Photorealistic Animation and Rendering '00, pages 13–20, New York, NY, USA, 2000. ACM.

[94] Roberto Lam, Jo ao Rodrigues, and J.M.Hans du Buf. Looking through the eye of the painter from the visual cortex and brightness perception to non-photorealistic. In Joaquim Jorge and Vaclav Skala, editors, *14th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision (WSCG'06)*, pages 147–154, Plzen, 2006. UNION Agency.

[95] Daniel L. Lau and Gonzalo R. Arce. *Modern Digital Halftoning, Second Edition (Signal Processing and Communications) (Hardcover)*. CRC; 2 edition (April 24, 2008), 2008.

[96] Hyunjun Lee, Sungtae Kwon, and Seungyong Lee. Real-time pencil rendering. In *Proceedings of the 4th international symposium on Non-photorealistic animation and rendering*, Non-Photorealistic Animation and Rendering '06, pages 37–45, New York, NY, USA, 2006. ACM.

[97] Hua Li and David Mould. Contrast-aware halftoning. *Computer Graphics Forum*, 29(2):273–280, 2010.

[98] Hua Li and David Mould. Artistic tessellations by growing curves. In *Proceedings of the 6th international symposium on Non-photorealistic animation and rendering*, Non-photorealistic animation and rendering '11, pages 125–134, New York, NY, USA, 2011. ACM.

[99] Hua Li and David Mould. Content-sensitive screening in black and white. In *International Conference on Computer Graphics Theory and Applications (GRAPP)*, pages 166–172. SciTePress, 2011.

[100] Hua Li and David Mould. Structure-preserving stippling by priority-based error diffusion. In *Proceedings of Graphics Interface 2011*, Graphics Interface (GI '11), pages 127–134. Canadian Human-Computer Communications Society, 2011.

[101] Xin Li. Edge-directed error diffusion halftoning. *IEEE Signal Processing Letters*, 13(11):688–690, November 2006.

[102] Yanxi Liu, Hagit Hel-Or, and Craig S. Kaplan. *Computational Symmetry in Computer Vision and Computer Graphics (Foundations and Trends(r) in Computer Graphics and Vision)*. Now Publishers Inc, 2010.

[103] Yu Liu, Olga Veksler, and Olivier Juan. Simulating classic mosaics with graph cuts. In *Proceedings of the 6th international conference on Energy minimization methods in computer vision and pattern recognition*, EMMCVPR'07, pages 55–70, Berlin, Heidelberg, 2007. Springer-Verlag.

[104] Yu Liu, Olga Veksler, and Olivier Juan. Generating classic mosaics with graph cuts. *Computer Graphics Forum*, 29(8):2387–2399, 2010.

[105] Paul J. Locher, Pieter Jan Stappers, and Kees Overbeeke. The role of balance as an organizing design principle underlying adults' compositional strategies for creating visual displays. *Acta Psychologica*, 99(2):141 – 161, 1998.

[106] Jorge Lopez-Moreno, Jorge Jimenez, Sunil Hadap, Erik Reinhard, Ken Anjyo, and Diego Gutierrez. Stylized depiction of images based on depth perception. In *Proceedings of the 8th International Symposium on Non-Photorealistic Animation and Rendering*, Non-Photorealistic Animation and Rendering '10, pages 109–118, New York, NY, USA, 2010. ACM.

[107] Cewu Lu, Li Xu, and Jiaya Jia. Combining sketch and tone for pencil drawing production. In *Proceedings of the Symposium on Non-Photorealistic Animation and Rendering*, Non-Photorealistic Animation and Rendering '12, pages 65–73, Aire-la-Ville, Switzerland, Switzerland, 2012. Eurographics Association.

[108] Peter J. Lu and Paul J. Steinhardt. Decagonal and quasi-crystalline tilings in medieval Islamic architecture. *Science*, 315(5815):1106–1110, Feburary 2007.

[109] Thomas Luft and Oliver Deussen. Interactive watercolor animations. In *13th Pacific Conference on Computer Graphics and Applications (PG'05)*, pages 7–9, Macau, October 12-14th 2005. Universität Konstanz Fachbereich Informatik und Informationswissenschaft. Fachbereich Informatik und Informationswissenschaft 2005.

[110] Thomas Luft and Oliver Deussen. Real-time watercolor illustrations of plants using a blurred depth test. In *Proceedings of the 4th international symposium on Non-photorealistic animation and rendering*, Non-photorealistic animation and rendering '06, pages 11–20, New York, NY, USA, 2006. ACM.

[111] Eric B. Lum and Kwan-Liu Ma. Non-photorealistic rendering using watercolor inspired textures and illumination. In *Proceedings of the 9th Pacific Conference on Computer Graphics and Applications*, Pacific Graphics (PG '01), pages 322–331, Washington, DC, USA, 2001. IEEE Computer Society.

[112] Aditi Majumder and M. Gopi. Hardware accelerated real time charcoal rendering. In *Proceedings of the 2nd international symposium on Non-photorealistic animation and rendering*, Non-Photorealistic Animation and Rendering '02, pages 59–66, New York, NY, USA, 2002. ACM.

[113] Regan L. Mandryk, David Mould, and Hua Li. Evaluation of emotional response to non-photorealistic images. In *Non-Photorealistic Animation and Rendering '11*, pages 7–16. ACM, 2011.

[114] James L. Mannos and David J. Sakrison. The effects of a visual fidelity criterion on the encoding of images. *IEEE Transactions on Information Theory*, IT-20(4):525–536, 1974.

[115] Xiaoyang Mao, Yoshiyasu Nagasaka, and Atsumi Imamiya. Automatic generation of pencil drawing from 2D images using line integral convolution. In *7th International Conference on Computer Aided Design and Computer Graphics (CAD/GRAPHICS'01)*, pages 240–248. IAP, August 2001.

[116] Gabriel Marcu and Satoshi Abe. Halftoning by back error compensation. In *Proceedings, NIP12: International Conference on Digital Printing Technologies*, pages 132–135, Springfield, VA, 1996. IS & T.

[117] Domingo Martín, Germán Arroyo, M. Victoria Luzón, and Tobias Isenberg. Example-based stippling using a scale-dependent grayscale process. In *Proceedings of the 8th International Symposium on Non-Photorealistic Animation and Rendering*, Non-Photorealistic Animation and Rendering '10, pages 51–61, New York, NY, USA, 2010. ACM.

[118] Domingo Martín, Germán Arroyo, M. Victoria Luzón, and Tobias Isenberg. Scale-dependent and example-based grayscale stippling. *Computers & Graphics*, 35(1):160 – 174, 2011.

[119] Krešimir Matković, László Neumann, Attila Neumann, Thomas Psik, and Werner Purgathofer. Global contrast factor - a new approach to image contrast. In *Proceedings of the First Eurographics conference on Computational Aesthetics in Graphics, Visualization and Imaging*, Computational Aesthetics'05, pages 159–167, Aire-la-Ville, Switzerland, Switzerland, 2005. Eurographics Association.

[120] Barbara J. Meier. Painterly rendering for animation. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, SIGGRAPH '96, pages 477–484, New York, NY, USA, 1996. ACM.

[121] Theophano Mitsa and Kevin J. Parker. Digital halftoning using a blue noise mask. In *Proceedings of SPIE*, volume 1452 of *Image Processing Algorithms and Techniques II*, pages 47–56. SPIE, June 1991.

[122] David Mould. A stained glass image filter. In *Proceedings of the 14th Eurographics workshop on Rendering*, Eurographics Workshop on Rendering (EGRW '03), pages 20–25, Aire-la-Ville, Switzerland, Switzerland, 2003. Eurographics Association.

[123] David Mould. Stipple placement using distance in a weighted graph. In *Proceedings of the Third Eurographics conference on Computational Aesthetics in Graphics, Visualization and Imaging*, Computational Aesthetics'07, pages 45–52, Aire-la-Ville, Switzerland, Switzerland, 2007. Eurographics Association.

[124] David Mould. Texture-preserving abstraction. In *Proceedings of the Symposium on Non-Photorealistic Animation and Rendering*, Non-Photorealistic Animation and Rendering '12, pages 75–82, Aire-la-Ville, Switzerland, Switzerland, 2012. Eurographics Association.

[125] David Mould and Kevin Grant. Stylized black and white images from photographs. In *Proceedings of the 6th international symposium on Non-photorealistic animation and rendering*, Non-photorealistic animation and rendering '08, pages 49–58, New York, NY, USA, 2008. ACM.

[126] David Mould, Regan L. Mandryk, and Hua Li. Emotional response and visual attention to non-photorealistic images. *Computers & Graphics*, 36(6):658–672, 2012.

[127] Michael Neff and Eugene Fiume. A visual model for blast waves and fracture. In *Proceedings of the 1999 conference on Graphics interface '99*, pages 193–202, San Francisco, CA, USA, 1999. Morgan Kaufmann Publishers Inc.

[128] U. Neisser. *Cognitive Psychology*. Addison-Wesley ProfesAppleton-Century-Crofts, 1967.

[129] László Neumann, Kresimir Matkovic, and Werner Purgathofer. Global contrast factor. In A. Hast E. Carling, editor, *Seminar on Computer Graphics and Visualization*, pages 108–119. University of Gavle, November 2002.

[130] Robert Ogniewicz and M. Ilg. Voronoi skeletons: Theory and applications. In *Proceedings of Conference on Computer Vision and Pattern Recognition*, pages 63–69. IEEE Computer Society, 1992.

[131] Atsuyuki Okabe, Barry Boots, and Kokichi Sugihara. *Spatial tessellations: concepts and applications of Voronoi diagrams*. John Wiley & Sons, Inc., New York, NY, USA, 1992.

[132] Jeff Orchard and Craig S. Kaplan. Cut-out image mosaics. In *Proceedings of the 6th international symposium on Non-photorealistic animation and rendering*, Non-photorealistic animation and rendering '08, pages 79–87, New York, NY, USA, 2008. ACM.

[133] Victor Ostromoukhov. Artistic halftoning: between technology and art. In *Proceedings of SPIE*, volume 3963 of *Color Imaging: Device-Independent Color, Color Hardcopy, and Graphic Arts V*, pages 489–509, San Jose, CA, December 1999. SPIE.

[134] Victor Ostromoukhov. A simple and efficient error-diffusion algorithm. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '01, pages 567–572, New York, NY, USA, 2001. ACM.

[135] Victor Ostromoukhov and Roger D. Hersch. Artistic screening. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, SIGGRAPH '95, pages 219–228, New York, NY, USA, 1995. ACM.

[136] Victor Ostromoukhov and Roger D. Hersch. Halftoning by rotating non-bayer dispersed dither arrays. In *Proceedings of SPIE*, volume 2411, pages 180–197. SPIE, 1995.

[137] Victor Ostromoukhov and Roger D. Hersch. Multi-color and artistic dithering. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '99, pages 425–432, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co.

[138] Victor Ostromoukhov, Roger D. Hersch, and Isaac Amidror. Rotated dispersed dither: a new technique for digital halftoning. In *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, SIGGRAPH '94, pages 123–130, New York, NY, USA, 1994. ACM.

[139] Wai-Man Pang, Yingge Qu, Tien-Tsin Wong, Daniel Cohen-Or, and Pheng-Ann Heng. Structure-aware halftoning. *ACM Transactions on Graphics*, 27(3):89:1–89:8, August 2008.

[140] J. R. Pomerantz and M Kubovy. Theoretical approaches to perceptual organization: Simplicity and likelihood principles. *In K. R. Boff, L. Kaufman, & J. P. Thomas (Eds.), Handbook of Perception and Human Performance: Vol. 2. Cognitive Processes and Performance*, pages 36–1–36–46, 1986.

[141] Thierry Pudet. Real time fitting of hand-sketched pressure brushstrokes. *Computer Graphics Forum*, 13(3):205–220, 1994.

[142] Yingge Qu, Wai-Man Pang, Tien-Tsin Wong, and Pheng-Ann Heng. Richness-preserving manga screening. *ACM Transactions on Graphics*, 27(5):155:1–155:8, December 2008.

[143] Paul Rademacher, Jed Lengyel, Edward Cutrell, and Turner Whitted. Measuring the perception of visual realism in images. In StevenJ. Gortler and Karol Myszkowski, editors, *Rendering Techniques 2001*, Eurographics, pages 235–247. Springer Vienna, 2001.

[144] J. Rangel-Mondragon and S.J. Abas. Computer generation of Penrose tilings. *Computer Graphics Forum*, 7(1):29–37, 1988.

[145] William T. Reeves. Particle systems — technique for modeling a class of fuzzy objects. *ACM Transactions on Graphics*, 2(2):91–108, April 1983.

[146] David Regan. *Human Perception of Objects: Early Visual Processing of Spatial Form Defined by Luminance, Color, Texture, Motion, and Binocular Disparity.* Sinauer Associates; 1st edition, March 2000.

[147] David Regan. *Human Percetpion of Objects: Early Visual Processing of Spatial Form Defined by Luminance, Color, Texture, Motion, and Binocular Disparity.* Sinauer Associates; 1st edition, Sutherland, Massachusetts, USA, 2000.

[148] Sheena Rogers. Perceiving pictorial space. *W. Epstein & S.Rogers (Eds.), Perception of space and motion*, pages 119–163, 1995.

[149] Paul Rosin and John Collomosse. *Image and Video-Based Artistic Stylisation.* Springer, 2013.

[150] Paul L. Rosin and Yu-Kun Lai. Towards artistic minimal rendering. In *Proceedings of the 8th International Symposium on Non-Photorealistic Animation and Rendering*, Non-Photorealistic Animation and Rendering '10, pages 119–127, New York, NY, USA, 2010. ACM.

[151] H. Rushmeier, G. Ward, C. Piatko, P. Sanders, and B. Rust. Comparing real and synthetic images: Some ideas about metrics. In PatrickM. Hanrahan and Werner Purgathofer, editors, *Eurographics Rendering Workshop 1995*, Eurographics, pages 82–91. Springer Vienna, 1995.

[152] Michael P. Salisbury, Sean E. Anderson, Ronen Barzel, and David H. Salesin. Interactive pen-and-ink illustration. In *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, SIGGRAPH '94, pages 101–108, New York, NY, USA, 1994. ACM.

[153] Michael P. Salisbury, Michael T. Wong, John F. Hughes, and David H. Salesin. Orientable textures for image-based pen-and-ink illustration. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '97, pages 401–406, New York, NY, USA, 1997. ACM Press/Addison-Wesley Publishing Co.

[154] Mike Salisbury, Corin Anderson, Dani Lischinski, and David H. Salesin. Scale-dependent reproduction of pen-and-ink illustrations. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, SIGGRAPH '96, pages 461–468, New York, NY, USA, 1996. ACM.

[155] Dario D. Salvucci and John R. Anderson. Automated eye-movement protocol analysis. *Human-Computer Interaction*, 16(1):39–86, March 2001.

[156] Anthony Santella. *The Art of Seeing: Visual Perception in Design and Evaluation of Non-Photorealistic Rendering*. PhD thesis, Rutgers University, New Brunswick, New Jersey, USA, May 2005.

[157] Thomas Scheermesser and Olof Bryngdahl. Control of texture in image halftoning. *Journal of the Optical Society of America A (JOSA A)*, 13(8):1645–1652, 1996.

[158] Stefan Schlechtweg, Tobias Germer, and Thomas Strothotte. RenderBots – multi agent systems for direct image generation. *Computer Graphics Forum*, 24(2):137–148, 2005.

[159] Christian Schmaltz, Pascal Gwosdek, and Joachim Weickert. Multi-class anisotropic electrostatic halftoning. *Computer Graphics Forum*, 31(6):1924–1935, 2012.

[160] Jutta Schumann, Thomas Strothotte, Stefan Laser, and Andreas Raab. Assessing the effect of non-photorealistic rendered images in CAD. In *Proceedings of the SIGCHI conference on Human factors in computing systems: common ground*, CHI '96, pages 35–41, New York, NY, USA, 1996. ACM.

[161] Adrian Secord. Weighted Voronoi stippling. In *Proceedings of the 2nd international symposium on Non-photorealistic animation and rendering*, Non-photorealistic animation and rendering '02, pages 37–43, New York, NY, USA, 2002. ACM.

[162] Ojaswa Sharma, Darka Mioc, and Francois Anton. Voronoi diagram based automated skeleton extraction from colour scanned maps. In *Proceedings of the 3rd International Symposium on Voronoi Diagrams in Science and Engineering*, pages 186–195. IEEE Computer Society, 2006.

[163] Kaleigh Smith, Yunjun Liu, and Allison Klein. Animosaics. In *Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*, Symposium on Computer Animation (SCA '05), pages 201–208, New York, NY, USA, 2005. ACM.

[164] Minjung Son, Henry Kang, Yunjin Lee, and Seungyong Lee. Abstract line drawings from 2D images. In *Proceedings of the 15th Pacific Conference on Computer Graphics and Applications*, pages 333–342, Washington, DC, USA, 2007. IEEE Computer Society.

[165] Minjung Son, Yunjin Lee, Henry Kang, and Seungyong Lee. Structure grid for directional stippling. *Graphical Models*, 73(3):74 – 87, 2011.

[166] Milan Sonka, Vaclav Hlavac, and Roger Boyle. *Image Processing: Analysis and Machine Vision*. CL-Engineering; 2 edition, 1998.

[167] Mario Costa Sousa and John W. Buchanan. Computer-generated graphite pencil rendering of 3d polygonal models. *Computer Graphics Forum*, 18(3):195–208, 1999.

[168] Mario Costa Sousa and John W. Buchanan. Observational models of graphite pencil materials. *Computer Graphics Forum*, 19(1):27–49, March 2000.

[169] Lisa Streit and John Buchanan. Importance driven halftoning. *Computer Graphics Forum*, 17(3):207–217, 1998.

[170] Lisa Streit, Oleg Veryovka, and John Buchanan. Non-photorealistic rendering using an adaptive halftoning technique. In *Computer Graphics Symposium (SKIGRAPH 1999)*, pages 37–45. University of Calgary, 1999.

[171] Thomas Strothotte and Stefan Schlechtweg. *Non-Photorealistic Computer Graphics: Modeling, Rendering, and Animation (The Morgan Kaufmann Series in Computer Graphics)*. Morgan Kaufmann; 1st edition, 2002.

[172] Ivan E. Sutherland. Sketch pad a man-machine graphical communication system. In *Proceedings of the SHARE design automation workshop*, DAC '64, pages 6.329–6.346, New York, NY, USA, 1964. ACM.

[173] Jared Tarbell. Substrate. `http://www.complexification.net/`, 2012.

[174] Robert Ulichney. *Digital Halftoning*. MIT Press, 1987.

[175] Robert A. Ulichney. One-dimensional dithering. In *Proceedings of SPIE*, volume 3409, pages 204–214. SPIE, 1998.

[176] David Vanderhaeghe, Pascal Barla, Joëlle Thollot, and François Sillion. Dynamic point distribution for stroke-based rendering. In *Proceedings of the 18th Eurographics conference on Rendering Techniques*, Eurographics Symposium on Rendering '07, pages 139–146, Aire-la-Ville, Switzerland, Switzerland, 2007. Eurographics Association.

[177] Luiz Velho and Jonas de Miranda Gomes. Digital halftoning with space filling curves. In *Proceedings of the 18th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '91, pages 81–90, New York, NY, USA, 1991. ACM.

[178] Oleg Verevka and John W. Buchanan. Halftoning with image-based dither screens. In *In Proceedings of the 1999 conference on Graphics interface'99*, pages 167–174, Kingston, Ontario, Canada, 1999. Morgan Kaufmann Publishers Inc.

[179] Oleg Veryovka and John Buchanan. Comprehensive halftoning of 3D scenes. *Computer Graphics Forum (Eurographics)*, 18(3):13–22, 1999.

[180] Oleg Veryovka and John Buchanan. Texture-based dither matrices. *Computer Graphics Forum*, 19(1):51–64, 2000.

[181] Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

[182] Severin Weissbach and Olof Bryngdahl. Control of halftone texture by error diffusion. *Optics Communications*, 103(3-4):174 – 180, 1993.

[183] Max Wertheimer. Laws of organization in perceptual forms. *Psychologische Forschung*, 3(4):301 – 350, 1923.

[184] Brett Wilson and Kwan-Liu Ma. Rendering complexity in computer-generated pen-and-ink illustrations. In *Proceedings of the 3rd international symposium on Non-photorealistic animation and rendering*, Non-Photorealistic Animation and Rendering '04, pages 129–137, New York, NY, USA, 2004. ACM.

[185] Georges Winkenbach and David H. Salesin. Computer-generated pen-and-ink illustration. In *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, SIGGRAPH '94, pages 91–100, New York, NY, USA, 1994. ACM.

[186] Georges Winkenbach and David H. Salesin. Rendering parametric surfaces in pen and ink. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, SIGGRAPH '96, pages 469–476, New York, NY, USA, 1996. ACM.

[187] Holger Winnemöller. *Perceptually-motivated non-photorealistic graphics*. PhD thesis, Evanston, IL, USA, 2006. Northwestern University, AAI3230188.

[188] Holger Winnemöller, Jan Eric Kyprianidis, and Sven C. Olsen. XDoG: An extended difference-of-gaussians compendium including advanced image stylization. *Computers & Graphics*, 36(6):740 – 753, 2012.

[189] WiseGEEK. Contrast effect. online, 2003. `http://www.wisegeek.com/what-is-a-contrast-effect.htm`.

[190] J M Wolfe. Guided search 2.0: A revised model of visual search. *Psychonomic Bulletin & Review*, 1(2):202–238, 1994.

[191] Michael T. Wong, Douglas E. Zongker, and David H. Salesin. Computer-generated floral ornament. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '98, pages 423–434, New York, NY, USA, 1998. ACM.

[192] Huisi Wu, Tien-Tsin Wong, and Pheng-Ann Heng. Parallel structure-aware halftoning. *Multimedia Tools and Applications*, 67(3):529–547, 2013.

[193] Jie Xu and Craig S. Kaplan. Artistic thresholding. In *Proceedings of the 6th international symposium on Non-photorealistic animation and rendering*, Non-photorealistic animation and rendering '08, pages 39–47, New York, NY, USA, 2008. ACM.

[194] Ling Xu and David Mould. Magnetic curves: curvature-controlled aesthetic curves using magnetic fields. In *Proceedings of the Fifth Eurographics conference on Computational Aesthetics in Graphics, Visualization and Imaging*, Computational Aesthetics '09, pages 1–8, Aire-la-Ville, Switzerland, Switzerland, 2009. Eurographics Association.

[195] Shigefumi Yamamoto, Xiaoyang Mao, and Atsumi Imamiya. Enhanced lic pencil filter. In *Proceedings of the International Conference on Computer Graphics, Imaging and Visualization*, pages 251–256, Washington, DC, USA, 2004. IEEE Computer Society.

[196] Chuan-Kai Yang and Hui-Lin Yang. Realization of Seurat's pointillism via non-photorealistic rendering. *The Visual Computer*, 24(5):303–322, 2008.

[197] Ruiko Yano and Yasushi Yamaguchi. Texture screening method for fast pencil rendering. *Journal for Geometry and Graphics*, 9(2):191–200, 2005.

[198] Mi You, Taekwon Jang, Seunghoon Cha, Jihwan Kim, and Junyong Noh. Realistic paint simulation based on fluidity, diffusion, and absorption. *Journal of Visualization and Computer Animation*, 24(3-4):297–306, 2013.

[199] Johannes Zander, Tobias Isenberg, Stefan Schlechtweg, and Thomas Strothotte. High quality hatching. *Computer Graphics Forum*, 23(3):421–430, 2004.

[200] Robert C. Zeleznik, Kenneth P. Herndon, and John F. Hughes. SKETCH: an interface for sketching 3D scenes. In *SIGGRAPH '96: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 163–170, New York, NY, USA, 1996. ACM.

[201] Robert C. Zeleznik, Kenneth P. Herndon, and John F. Hughes. Sketch: an interface for sketching 3D scenes. In *ACM SIGGRAPH 2007 courses*, SIGGRAPH '07, New York, NY, USA, 2007. ACM.

[202] Kun Zeng, Mingtian Zhao, Caiming Xiong, and Song-Chun Zhu. From image parsing to painterly rendering. *ACM Transactions on Graphics*, 29(1):2:1–2:11, December 2009.

[203] Mingtian Zhao and Song-Chun Zhu. Sisley the abstract painter. In *Proceedings of the 8th International Symposium on Non-Photorealistic Animation and Rendering*, Non-Photorealistic Animation and Rendering '10, pages 99–107, New York, NY, USA, 2010. ACM.

[204] Mingtian Zhao and Song-Chun Zhu. Customizing painterly rendering styles using stroke processes. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Non-Photorealistic Animation and Rendering*, Non-Photorealistic Animation and Rendering '11, pages 137–146, New York, NY, USA, 2011. ACM.

[205] Bingfeng Zhou and Xifeng Fang. Improving mid-tone quality of variable-coefficient error diffusion using threshold modulation. *ACM Transactions on Graphics*, 22(3):437–444, July 2003.

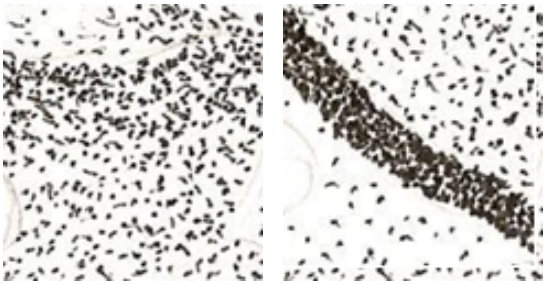
Appendices

Appendix A

Perception and Art

A.1 Perceptual Assistance

Visual perception is defined by Wertheimer [183] as "a primary mental ability to organize human sensations". The concept of perceptual assistance refers to employing perceptual characteristics in algorithmic design to help the brain in performing its visual task during the perceptual experience. The human perceptual system under the right conditions can be interrupted by saliencies, such as the sudden onset of a stimulus, motion, and sharp contrasts, which can capture visual attention [146]. For example, we are more attracted by the strong contrast between density for edges in Figure A.1 (b) than in (a). This illustration shows how a contrast-enhanced distribution for expressing an edge with dots gains more attention than a distribution with less contrast. In Figure A.1 (a), it is difficult for a viewer to detect the edge while the one in Figure A.1 (b) is easily recognized. The contrast between distribution in Figure A.1 (b) can clearly display edges. The key idea from perceptual assistance is to make placement cater to perceptual rules. More specifically, it means, distributing primitives through space in a way to show certain perceptual effects, such as contrast, similarity, and geometrical balance, can help our perception attain the sensation of visual experience. Next we discuss three important perceptual features: constancy, contrast effects, and grouping.



(a) Without perceptual assistance

(b) With perceptual assistance

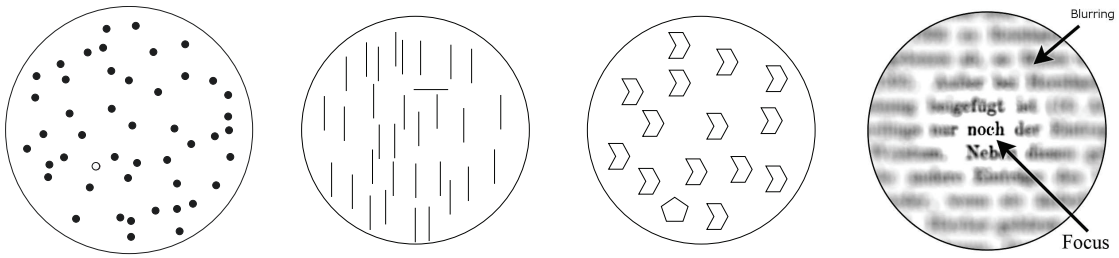
Figure A.1: Distributions without and with perceptual assistance.

A.1.1 Perceptual Constancy

Perceptual constancy is the ability to recognise the same object from varying sensory inputs. For example, when an object is positioned in a different view, humans can easily discern that it is the same object. There are various types of constancy, including brightness constancy, size constancy, and shape constancy. Humans have the ability of perceptual constancy using which the human perception system can extract invariant properties from extrinsic properties. Constancy has an important impact on art. A typical example in art is Realism, in which the goal is to create images that look like real pictures taken by camera. A familiar scene depicted in realistic art in viewer's mind is because of perceptual constancy. Hence, in order to convey information meaningfully, NPR algorithms should find a way to present constancy in the results. Correctly extracting intrinsic properties of objects, such as size, color, and shape, can help depict the main information efficiently. As a consequence, extracting important properties of objects has been an essential task in NPR research for a while.

A.1.2 Contrast Effect

The contrast effect is a fundamental principle of perception, related to the enhancement or diminishment of perception. Contrast effect is defined as "the phenomenon where people perceive greater or lesser differences than are actually present as a result of prior or simultaneous exposure to something with similar base characteristics, but different key qualities" [189].As an object has different attributes, there are various types of contrast, for example, focused and blurred, random and ordered. Figure A.2 shows a collection of examples for four types of contrast. A black dot with a hollow is easily detected when it is surrounded by dark solid dots, which is because of color difference; a horizontal line is obviously seen circled by vertical lines, which is because of direction difference; a concave is not difficult to find in a set of convex because of the shape difference; a focused letter is clearly read from blurring words because of the blurriness difference. The placement of primitives relies on the contrast between neighboring contexts to expressively transmit information. The illustration in Figure A.2 also informs us visual effects from dots, lines, shapes are quite different as each primitive can be varied visually from its attributes. There are different types



(a) (b) (c) (d)

Figure A.2: Attention from different contrast types. (a) Color contrast; (b) direction contrast; (c) shape contrast; (d) blurriness contrast.

of contrast according to the difference between primitive attributes. Artists express contrast effects a lot in their artistic work such as black and white, scratchboard, heightening. Strong visual contrasts capture people's attention. Contrast effects have been researched in many areas such as image processing for image enhancement. However, a fairly small number of researchers [15, 23, 150] addressed NPR issues related to perceptual contrast in placement. DeCarlo and Rusinkiewicz [23] used contrast to highlight suggestive contours to convey shape cues. The small indication of very bright regions enhancing contrast conveyed the final effect vividly.

A.1.3 Gestalt Laws of Grouping

The principles of grouping are a set of principles in psychology, following the observation that humans naturally perceive objects as organized patterns and objects. More specifically, the perceptual grouping is also called Gestalt laws of grouping, where Gestalt roughly means "whole", "form", or "pattern" in translation. The bits of sensory information that we receive from individual visual elements are not continuous, but disconnected. The brain can organize those bits of information into a meaningful whole. Six common principles are the following: proximity, similarity, closure, good continuation, symmetry, and simplicity, as illustrated in Figure A.3. Basically, a viewer tends to order his experience in a manner that is regular, orderly, symmetric, and simple. Symmetry, for example, is a commonly-known aesthetic feature applied by designers and artists in architectural design and artwork such as Islamic patterns and Chinese paper-cuts. In the field of NPR, Wong et al. [191] successfully

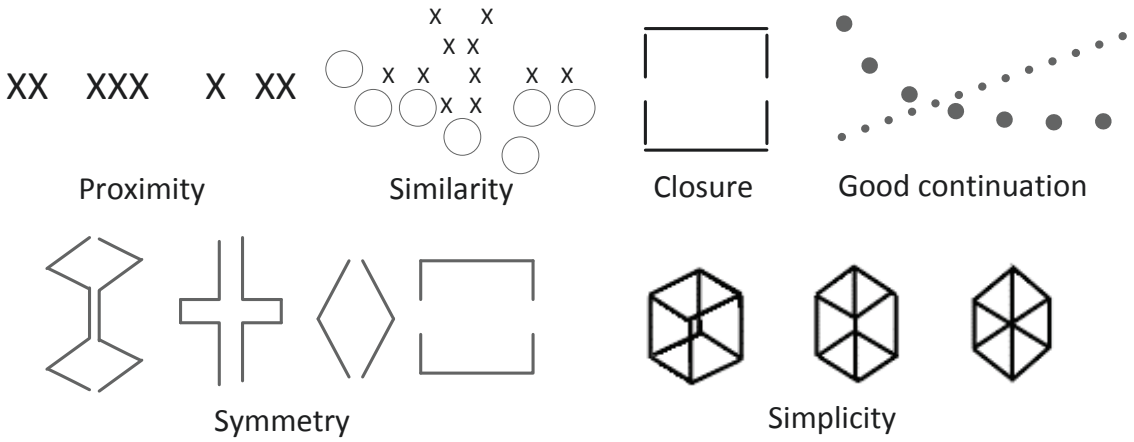


Figure A.3: Six Gestalt laws of grouping.

composed symmetric placement for computer-generated floral ornament. Kaplan and Salesin [79, 80, 81] presented an extensive series of work for tilings and patterns, used for Escherization and Islamic patterns. One strategy Kaplan and Salesin employed is to present symmetry and similarity in design.

A.2 Good Composition

The quality of abstraction and composition of abstract patterns to some degree relies on how the primitives are placed. Composition in visual art is the placement or arrangement of primitives. In order to have good design, cooperation between creative thoughts and perceptual organization is indispensable. So a picture created by an artist is not a random slice of marks, but is designed [148]. Goldstein wrote that a "good" composition maintains three perceptual effects [30]: (a) it helps achieve perceptual clarity; (b) it creates a feeling of harmony or balance; and (c) it leads the viewer's attention to various areas of the picture.

We first would like to achieve perceptual clarity in visual appearance. For abstracting objects in an image, a placement of primitives should depict the properties of objects, such as size, shape, and structure, in order to help viewers in perceptual constancy. Good composition avoids pictorial "mistakes" or confusions. For example, Figure A.4 shows a house drawing that has a false attachment of the edges due to prolonging the strokes (circled in blue) accidently. This false stroke is unwanted.

Figure A.4: A house with an unwanted edge, adapted from Flickr user dorywithserifs.

An incorrect placement of primitives makes a picture ineffective in passing information to a viewer, whereupon he loses interest. Showing the content correctly is an important goal to achieve perceptual clarity. Each visual primitive such as a dot, a stroke, a line, and a curve by itself cannot be meaningful. The visual effects stem from the context or the mutual interaction between individual primitives and their neighboring primitives. Increasing perceptual contrast can enlarge the difference between the properties and then ensures the strength of the figure-ground segregation, which makes pictures easier to read. Based on this, beyond perceptual correctness, to have perceptual clarity is to maintain contrast effects and hence to enhance the content.

Secondly, we would like to create aesthetic effects and balance in composition. Since the context in a distribution of visual primitives is important, creating aesthetic effects needs presenting the harmony and the balance among primitives — a commonly-known goal in composition [105]. The textbook "*Fundamentals of Art*" [32] defined balance as "a sense of equilibrium achieved through implied weight, attention, or attraction, by manipulating the visual primitives within an artwork and as the optimal equilibrium among all parts of the work". Therefore, balance is a psychological state created by the arrangement of pictorial primitives. Weights are spatially determined both by qualities of the objects and by their location in the picture.

The third perceptual effect is to direct attention. Perhaps more difficult, a good placement of primitives in art creation could direct attention to draw the viewer into the picture, which means good composition can influence where a viewer looks, and the order in which the objects are viewed.

A.3 Contrast between Properties of 2D Primitives

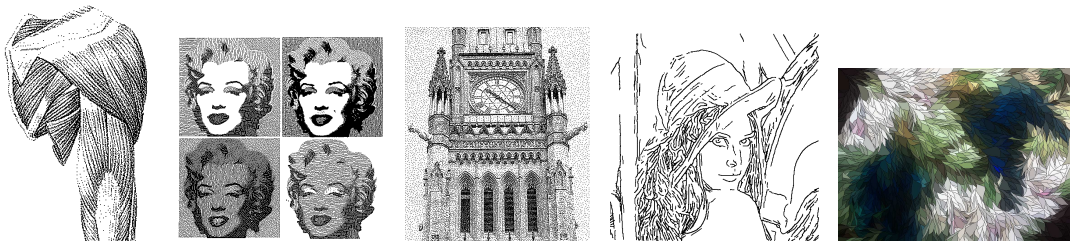
Differences between attributes of primitives cause different types of contrast. Figure A.5 shows different properties of those chosen primitives.

(a) A pixel (b) A dot (c) A line (d) A curve (e) A region

Figure A.5: Properties of primitives. (a) A pixel: position and color; (b) a dot: position, size, and color; (c) a line: thickness, position, color, length, and direction; (d) a curve: position, color, length, direction, and curvature; (e) a region closed by a set of curves.

A.3.1 Primitives

Primitives can be pixels, dots, lines, curves, and regions. A pixel is a resolution-dependent concept used in image space. Pixels in an image have position and color information. In binary images, each pixel is either black or white and the spacing between black pixels indicates the density of distribution directly. Large spacing indicates sparse distribution while small spacing suggests dense distribution. The contrast of a placement of pixels results from the spacing between black pixels. If a dot primitive has size information besides position and color information, the contrast can be the difference between the size values. A line primitive is straight and starts at a position with length along a direction. Thickness, direction, and length are three extra attributes to manipulate contrast for lines. A curve is formed by a set of lines and its curvature can be changed in creation; and thus the contrast can be shown in



(a) Halftoning (b) Screening (c) Stippling (d) Line Art (e) Mosaics

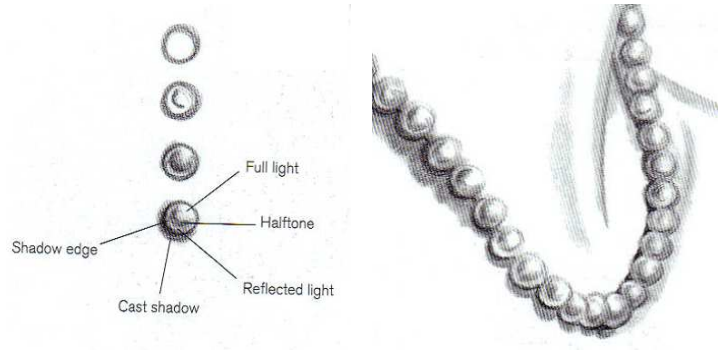
Figure A.6: Styles using different primitives. (a) pixel; (b) pixel; (c) dot; (d) line; (e) curve.

controlling the curvature. A region is a bounded area. A region boundary is formed by a series of connected curves. As a result, a region has more attributes than a pixel, a dot, a line, or a curve and provides more ways to display the difference between attributes. For example, the way of connecting the curves is a property of a region, which can be used to manage the contrast.

Previously, we proposed Prioritized CED and AT to create different styles using different primitives. Figure A.6 shows five different styles using different primitives, which are generated by the algorithms from this thesis. Figure (a), (b), (c), and (d), are produced by a contrast-aware framework to enhance the structure details. Figure (e) is created by a method employing similarity in placement of curves.

A.3.2 Depict Objects using Placement of Primitives

Usually significant shape information in a picture is provided by the object contours. Object surfaces under different lighting conditions provide gradual changes of tone. Shadows indicate depth and spatial relationships between objects. To express a picture, artists place strokes to capture the object contours, shading effects, or shadows. A common technique used by artists to create realistic drawing is shown in Figure A.7 (a), which demonstrates a five-element procedure for drawing an object. The five elements are shadow edge, halftone, full light, cast shadow, and reflected light. A real object like the pearl necklace in Figure A.7 (b) is composed of a few similar objects drawn individually. In the NPR literature, Hertzmann and Zorin [60]



(a) Steps for a pearl (b) A pearl necklace

Figure A.7: Realistic drawing techniques by five steps. Images are adapted from *Lifelike Drawing with Lee Hammond* [49].

presented computer-generated smooth hatching with artistic rules, which include to draw silhouettes, halftone, full light, and reflected light. The elements Hertzmann and Zorin used are similar, including halftone, full light, and reflected light, but not exact as the five elements Figure A.7 uses.

To express an edge using primitives, there are two ways: placing primitives on the featured edges and manipulating the attributes of primitives to show the contrast from both sides of the edges. Figure A.8 shows some ways of manipulation based on the difference in attributes for different primitives. In Figure A.8, we see examples of both strategies: placing pixels, lines, or curves on the edge; while taking advantage of contrast between some attribute to perceptually carve the edges. Intuitively, differences in more than one attribute produce stronger contrast, as shown in Figure A.8 (h). However, humans cannot pay attention to more than a few items simultaneously, so they are faced with the challenge of continuously integrating and prioritizing different influences [67], which are not fully explained by researchers in perception. Although, we can manipulate the visual appearance to communicate shading by adjusting attributes to present the change of tone, the arrangement of tone levels is subtle and continuous. We need to structure our placement of primitives for tone more carefully than for edges.

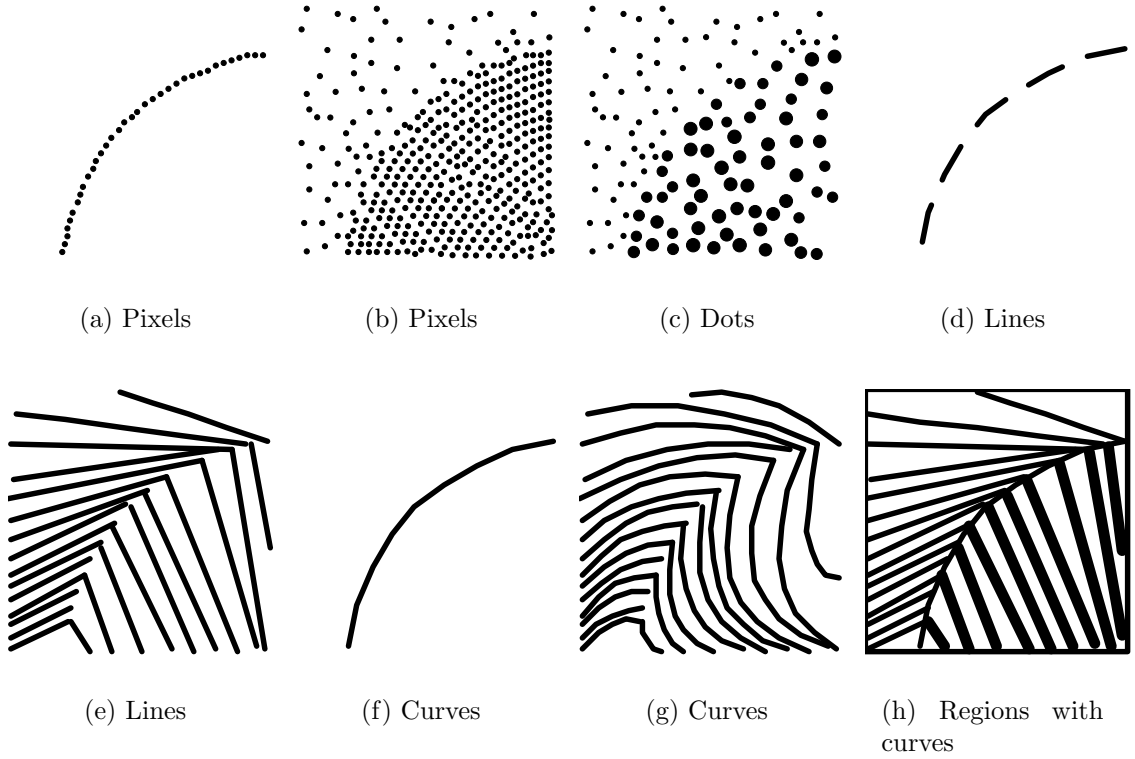


Figure A.8: To depict an edge, we show different ways of contrast between primitive attributes. (a) Pixels placed on the edge; (b) the difference between the distribution of pixels; (c) the difference between the distribution of dots; (d) lines placed on the edge; (e) the difference in line directions; (f) a curve placed on the edge; (g) the difference in curve curvatures; (h) the differences in line directions and a curve placed on the edge.