An Improved Error-Diffusion Approach for Generating Mesh Models of Images

by

Xiao Ma
B.Sc., Beijing University of Posts and Telecommunications, 2011

A Thesis Submitted in Partial Fulfillment of the
Requirements for the Degree of

MASTER OF APPLIED SCIENCE

in the Department of Electrical and Computer Engineering

An Improved Error-Diffusion Approach for Generating Mesh Models of Images

by

Xiao Ma
B.Sc., Beijing University of Posts and Telecommunications, 2011

Supervisory Committee

Dr. Michael Adams, Supervisor
(Department of Electrical and Computer Engineering)

Dr. Panajotis Agathoklis, Departmental Member
(Department of Electrical and Computer Engineering)

Supervisory Committee

Dr. Michael Adams, Supervisor
(Department of Electrical and Computer Engineering)

Dr. Panajotis Agathoklis, Departmental Member
(Department of Electrical and Computer Engineering)

ABSTRACT

Triangle mesh models of images are studied. Through exploration, a computational framework for mesh generation based on data-dependent triangulations (DDTs) and two specific mesh-generation methods derived from this framework are proposed.

In earlier work, Yang et al. proposed a highly-effective technique for generating triangle-mesh models of images, known as the error diffusion (ED) method. Unfortunately, the ED method, which chooses triangulation connectivity via a Delaunay triangulation, typically yields triangulations in which many (triangulation) edges crosscut image edges (i.e., discontinuities in the image), leading to increased approximation error. In this thesis, we propose a computational framework for mesh generation that modifies the ED method to use DDTs in conjunction with the Lawson local optimization procedure (LOP) and has several free parameters. Based on experimentation, we recommend two particular choices for these parameters, yielding two specific mesh-generation methods, known as MED1 and MED2, which make different trade offs between approximation quality and computational cost. Through the use of DDTs and the LOP, triangulation connectivity can be chosen optimally so as to minimize approximation error. As part of our work, two novel optimality criteria for the LOP are proposed, both of which are shown to outperform other well known criteria from the literature. Through experimental results, our MED1 and MED2 methods are shown to yield image approximations of substantially higher quality than those obtained with the ED method, at a relatively modest computational cost. For example, in terms of peak-signal-to-noise ratio, our MED1 and MED2 methods outperform the ED method, on average, by 3.26 and 3.81 dB, respectively.

Contents

List of Tables

List of Figures

List of Acronyms

ED	error diffusion
LOP	local optimization procedure
LLOP	look-ahead LOP
DDT	data-dependent triangulation
MSE	mean squared error
PSNR	peak-signal-to-noise ratio
MMSODD	maximum-magnitude second-order directional derivative
SE	squared error
ABN	angle between normals
JND	jump in normal derivatives
DLP	deviations from linear polynomials
DP	distances from planes
AMC	absolute mean curvature
GHH	Garland-Heckbert hybrid
SQSE	shape-quality-weighted SE
JNDSE	JND-weighted SE
ELSE	edge-length-weighted SE
MASE	minimum-angle-weighted SE

ACKNOWLEDGMENTS

This thesis would never have been written without the help and support from numerous people. I would like to take this opportunity to express my gratitude to a few of them in particular.

First and foremost, I would like to thank my supervisor, Dr. Michael Adams. Thank you for your mentorship and guidance throughout my graduate studies. Thank you for always being so nice and patient. From things as small as a text font to things as big as the research direction, you have always been rigorous and punctilious. Your hard working, your dedication to research and teaching have been contagious and have motivated me in my studies. It has been such a pleasure working with you.

Next, I would like to thank my supervisory member Dr. Pan Agathoklis. Thank you for being on my supervisory committee and spending time reviewing my thesis. I also want to express my gratitude to the course instructors during my graduate studies, Dr. Wu-Sheng Lu, Dr. Aaron Gulliver, Dr. Sue Whitesides, Dr. Bruce Kapron, and Dr. Peter Driessen. Thank you for your wonderful lectures and the efforts you have put in preparing for the courses.

Moreover, I would like to thank the staff members in the Department of Electrical and Computer Engineering, Janice Closson, Amy Rowe, Dan Mai, Moneca Bracken (retired), Kevin Jones, and Erik Laxdal. Thank you for your help in all aspects throughout my graduate studies.

Furthermore, I would like to thank my friends who have had an impact on me during my graduate studies, Qianqian Wang, Xinyu Fang, Guang Zeng, Jie He, Yuchen Wang, Xiao Feng, Leyuan Pan, Yongyu Dai, Le Liang, Yunlong Shao, and Ioana Sevcenco. Thank you for being my friends and I wish for our friendship to last beyond the duration of our time at UVic.

Last but not least, I want to thank my family. I want to thank my parents Yuening Ma and Jie Zhang first. Thank you for being such nice parents. Your unconditional love and support has been priceless and I feel so lucky to be your son. My grandparents, thank you for loving me for so many years, from my birth. Your love is invaluable to me. I wish you both healthy and happy everyday. My girlfriend, Mengyue Cai, thank you for showing up in my life. Your presence has been very important to me during my pursuit of the Master's degree.

DEDICATION

To my family.

Chapter 1

Introduction

1.1 Triangle Meshes for Image Representation

In real-world applications, images are typically nonstationary. Consequently, uniform sampling of images (such as with a truncated lattice) is usually far from optimal, with the sampling density inevitably being too high in some regions while too low in others. This has led to an interest in image representations based on nonuniform (i.e., content-adaptive) sampling. By choosing the sample points in a manner dependent on the image content, the number of samples can be greatly reduced. This smaller sample count can often be exploited in applications in order to reduce computational cost. Moreover, the sample data can often better capture the geometric structure inherent in images (such as image edges). In some applications, this can be exploited in order to obtain better quality results. Some applications in which nonuniform sampling has proven useful include: feature detection [1], pattern recognition [2], computer vision [3], restoration [4], tomographic reconstruction [5], filtering [6], interpolation [7, 8], and image/video coding [9, 10, 11, 12, 13, 14, 15].

Many general approaches to nonuniform sampling have been proposed to date. Some of the more popular approaches include inverse distance weighted methods [16, 17], radial basis function methods [16, 17], Voronoi and natural-neighbor interpolation methods [16], and finite-element methods [16, 17], including triangle meshes based on Delaunay triangulations [18, 19, 20, 21], constrained Delaunay triangulations [22], data-dependent triangulations [23, 24, 25, 26, 27, 28, 18, 29, 30], and geodesic triangulations [31]. Two excellent survey papers [16] and [17] present a good overview of the numerous general approaches to nonuniform sampling.

Among the numerous methods based on nonuniform sampling, one particularly effective approach is offered by triangle meshes. In this approach, the (nonuniformly chosen) sample points are triangulated, partitioning the image domain into triangular faces, and then an approximating function is constructed over each face of the triangulation. Triangle meshes are particularly effective for the following reasons. First, triangle meshes are well suited to capturing geometric structures in images (such as image edges). Moreover, triangle meshes have the potential to lead to memory-efficient nonuniform sampling methods, because a single triangle with only three sample points may cover a large area (with low variation) of an image, significantly reducing the amount of memory required for storing the image. Furthermore, triangle meshes greatly simplify the process of image interpolation. For example, the image approximation associated with a triangle mesh model can be easily obtained by combining the approximations constructed over each face of the triangulation. In addition, triangle meshes offer a way of handling image domains with arbitrary polygonal shapes.

Generally speaking, mesh-generation methods must address two problems: 1) the selection of the sample points; and 2) the selection of the connectivity of the triangulation of the sample points. Both of these problems are of interest in this thesis.

1.2 Historical Perspective

Considering the many advantages of triangle meshes as described above, numerous image representation methods based on triangle meshes have been proposed to date [23, 24, 25, 26, 27, 28, 18, 32, 29, 30, 33, 19, 20, 21, 34, 35, 36, 37, 10, 38, 39, 40]. Based on how the sample points are selected, the most popular mesh-generation methods can be classified into two categories: mesh-refinement schemes and mesh-simplification schemes, a few examples of which can be found in [30, 18, 41] and [35, 42], respectively. A mesh-refinement scheme works by starting with a very coarse triangulation (typically with the vertices being the extreme convex hull points), and adding one vertex at a time, with the new vertex chosen in a way so as to minimize some error measure, until the desired number of sample points is reached. In contrast, a mesh-simplification scheme starts with a refined triangulation (e.g., with the vertices being all the grid points in the image domain), and deletes one vertex at each step based on some error metric, until the vertices of the triangulation have been reduced to the desired size. Mesh-simplification schemes such as the adaptive thinning method of Demaret and Iske [35] are typically able to yield meshes with extremely high quality, but often demand high computational and memory requirements. On the other hand, mesh-refinement schemes such as the mesh-generation method of Rippa [41] are typically less expensive in terms of computation and memory cost, but they may result in lower quality meshes.

Regardless of how the sample points are selected, one key difference between the various triangle-mesh-based approaches is in how they select the triangulation connectivity (i.e., how the vertices of the triangulation are connected by edges). Among the many image representation methods that employ triangle meshes, the most common approach is to choose the connectivity by using a Delaunay triangulation [43]. In the case of Delaunay triangulations, the connectivity is determined solely by the set of sample points being triangulated. Delaunay triangulations maximize the minimum of all the interior angles of the triangles in the triangulation, thus avoiding sliver triangles to whatever extent is possible. This leads to the Delaunay triangulation being favored in approximation applications [44]. Although Delaunay triangulations are only guaranteed to be unique if no four points are cocircular, a unique Delaunay triangulation can be obtained easily by choosing an appropriate technique for handling degeneracies, such as preferred directions [45]. Examples of mesh-generation methods that are based on Delaunay triangulations are plentiful in literature, a few examples of which are [18, 19, 20, 21, 34, 35, 36].

Another approach to choosing the triangulation connectivity is offered by **data-dependent triangulations (DDTs)**. It is this particular approach that is of interest in this thesis. With a DDT, the triangulation connectivity can be chosen in an arbitrary manner, using information in the dataset from which the points to be triangulated were chosen. Since, unlike the Delaunay case, the connectivity of a DDT may be chosen arbitrarily, DDTs offer much greater flexibility, and have the potential to outperform their Delaunay counterparts if well chosen [41]. This said, however, connectivity selection is often a challenging task. Typically, optimization techniques are employed for this purpose, with the most common such technique, by far, being the **local optimization procedure (LOP)** of Lawson [46]. Examples of mesh-generation methods based on DDTs include [23, 24, 25, 26, 27, 28, 18, 32, 29, 30, 33, 38, 39, 40, 47, 48]. These approaches make heavy use of the LOP or variants thereof, such as the **look-ahead LOP (LLOP)** [29].

In [19], Yang et al. proposed a simple technique for generating triangle-mesh models of images, known as the **error-diffusion (ED)** method. The ED method selects the sample points in a way that they are distributed with a density approximately proportional to the maximum-magnitude second-order directional derivative of the original image, and triangulates the sample points using a Delaunay triangulation. Although this method has proven highly effective, it has the weakness that it often yields triangulations in which a significant number of (triangulation) edges crosscut image edges (i.e., discontinuities in the image), leading to a degradation in approximation quality. This weakness can be attributed to the fact that a Delaunay triangulation is employed by the ED method for choosing triangulation connectivity. The ED method serves as the motivation and foundation of the work in this thesis.

1.3 Overview and Contribution of the Thesis

In this thesis, we propose a computational framework for mesh generation that modifies the ED method to use DDTs in conjunction with the LOP. By using DDTs instead of Delaunay triangulations, we are able to better exploit triangulation connectivity in order to obtain the highest quality approximation. Using our computational framework, we derive two specific mesh-generation methods known as MED1 and MED2, which make different trade offs between approximation quality and computational cost. As we will show later, our MED1 and MED2 methods yield image approximations of substantially higher quality than those obtained with the ED method in terms of both **peak-signal-to-noise ratio (PSNR)** and subjective quality, at a relatively modest computational cost. As part of our work, we propose two novel optimality criteria for use with the LOP. Both of these criteria are shown to outperform numerous other well known criteria from the literature. In passing, we note that the work described herein has been partially presented in the author's papers [49, 50].

The remainder of this thesis is organized as follows. Chapter 2 provides some necessary background information to facilitate a better understanding of the work described herein. First, some basic notation and terminology are introduced. Then, some fundamentals in image processing and computational geometry are presented. After that, the triangle mesh models for image representation are discussed, followed by an introduction to a grid-point to face mapping scheme. Further, the ED method, on which our work is built, is presented. Lastly, a key algorithm used for selecting triangulation connectivity in the thesis, namely the LOP, is presented, along with a variant thereof, known as the LLOP.

Chapter 3 introduces our proposed approach and evaluates its performance. To begin, our proposed computational framework is introduced. The framework essentially modifies the ED method to use DDTs in conjunction with the LOP to better exploit triangulation connectivity in order to obtain the highest quality approximation. Then, we derive two specific mesh-generation methods using our framework, namely the MED1 and MED2 methods. This is done by first introducing several free parameters in our framework and the various choices for those parameters. After that, we study how different choices of those parameters will affect the mesh quality and advocate two particular choices of those parameters, leading to our two proposed methods. Furthermore, we evaluate our proposed MED1 and MED2 methods by comparing them to the ED scheme in terms of mesh quality and computational complexity. It is shown that our proposed MED1 and MED2 methods both yield image approximations of much higher quality than the ED scheme, at a relatively modest computational cost. For example, in terms of PSNR, our MED1 and MED2 methods outperform the ED method by 3.26 and 3.81 dB on average, respectively. It is also shown that the MED1 and MED2 methods make different trade offs between mesh quality and computational cost, as the MED2 method produces approximations of better quality, but at a higher computational cost. Lastly, some extra work we have done during the development of our framework is briefly discussed.

Chapter 4 concludes the thesis with a brief summary of the work presented herein and some suggestions for future research.

Appendix A provides a brief description of the software that is used to implement the computational framework proposed in the thesis and to collect experimental results. The software was fairly complex to develop but was designed to be user-friendly. Some instructions on how to use the software are also provided in this appendix.

Chapter 2

Preliminaries

2.1 Overview

In this chapter, we provide some background necessary to the understanding of the work presented in this thesis. To begin, we introduce the notation and terminology used in the remainder of the thesis. Then, some fundamentals in image processing and computational geometry are presented. After that, the triangle mesh models for image representation are discussed, followed by an introduction to the grid-point to face mapping scheme. Further, we present the ED scheme which serves as the foundation of our work. We conclude this chapter by introducing a key algorithm used in this thesis, namely the LOP, and a variant thereof, known as the LLOP.

2.2 Notation and Terminology

Before proceeding further, a brief digression is in order to introduce some basic notation and terminology employed throughout the thesis. The sets of integers and real numbers are denoted $\mathbb{Z}$ and $\mathbb{R}$, respectively. The cardinality of set S is denoted $|S|$. For a vector $v = (v_1, v_2, ..., v_n)$, the 2-norm of v is denoted $\|v\|$, and defined as

$$\|v\| = \sqrt{v_1^2 + v_2^2 + \ldots + v_n^2}.$$

2.3 Image Processing

Binomial Filter. Binomial filters [51] are simple and efficient structures that approximate Gaussian filtering based on binomial coefficients. One attractive property of binomial filters is that they do not require multiplications, which potentially benefits applications in terms of computational complexity. Because of their simplicity and efficiency, binomial filters are often used as lowpass filters for smoothing in image processing [52].

The transfer function of a one-dimensional (1-D) n-th order binomial filter (with zero phase and unity DC gain), is denoted as $H_n(z)$ and given by

$$H_n(z) = z^{\frac{(n-1)}{2}} \left(\frac{1}{2} + \frac{1}{2}z^{-1}\right)^{n-1},$$

where n is odd. For example, the nonzero coefficients of the impulse response of a 1-D third-order binomial filter are $[\frac{1}{4}\ \frac{1}{2}\ \frac{1}{4}]$. A 2-D binomial filter can be computed as the tensor product of two 1-D binomial filters. For example, the nonzero coefficients of the impulse response of a 2-D third-order binomial filter are $\begin{bmatrix} \frac{1}{16} & \frac{1}{8} & \frac{1}{16} \\ \frac{1}{8} & \frac{1}{4} & \frac{1}{8} \\ \frac{1}{16} & \frac{1}{8} & \frac{1}{16} \end{bmatrix}$.

MMSODD. For a function f defined on $\mathbb{R}^2$, the **maximum-magnitude second-order directional derivative (MMSODD)** $\tilde{m}$ of f is given by [19, Corollary 1 and Equation 12]

$$\tilde{m}(x,y) = \max\{|\alpha(x,y) + \beta(x,y)|, |\alpha(x,y) - \beta(x,y)|\}, \tag{2.1}$$

where

$$\alpha(x,y) = \frac{1}{2}\left[\frac{\partial^2}{\partial x^2} f(x,y) + \frac{\partial^2}{\partial y^2} f(x,y)\right]$$

and

$$\beta(x,y) = \sqrt{\frac{1}{4}\left[\frac{\partial^2}{\partial x^2} f(x,y) - \frac{\partial^2}{\partial y^2} f(x,y)\right]^2 + \left[\frac{\partial^2}{\partial x \partial y} f(x,y)\right]}.$$

The partial-derivative operators in the preceding equation are formed from the tensor product of 1-D derivative operators, where the discrete-time approximations of the 1-D first- and second-order derivative operators are computed using the filters with transfer functions $\frac{1}{2}z - \frac{1}{2}z^{-1}$ and $z - 2 + z^{-1}$, respectively. The MMSODD has a double response to image edges, with the maxima being attained just to each side of image edges. The MMSODD is of great importance to our work in this thesis, as we shall see later.

Skewness. For a function f defined on $\mathbb{R}^2$, the ***skewness*** $\tilde{s}$ of f, evaluated on a 3×3 window $\mathcal{M}$ centered at point (x,y), is given by [10, Equation 1]

$$\tilde{s}(x,y) = \frac{1}{9} \sum_{(x',y') \in \mathcal{M}_{x,y}} (f(x',y') - \mu(x,y))^3, \tag{2.2}$$

where

$$\mu(x,y) = \frac{1}{9} \sum_{(x',y') \in \mathcal{M}_{x,y}} f(x',y')$$

and

$$
\begin{aligned}
\mathcal{M} = \{&(x-1,y-1),(x,y-1),(x+1,y-1),\\
&(x-1,y),(x,y),(x+1,y),\\
&(x-1,y+1),(x,y+1),(x+1,y+1)\}.
\end{aligned}
$$

The skewness is a nonzero value except in uniform areas or in the case that $\mathcal{M}$ is centered exactly on an edge, and the absolute value of the skewness is higher in regions with rapid changes in an image.

2.4 Computational Geometry

In what follows, we introduce a few fundamental concepts from computational geometry such as the notions of a triangulation and a Delaunay triangulation. In order to define the concept of triangulation, we need to first introduce the concepts of convex set and convex hull.

Definition 2.1. *(Convex set). A set P of points in $\mathbb{R}^2$ is convex if for every pair of points $a,b \in P$, every point on the line segment that joins a and b is also in P.*

The definition of convex set is illustrated in Figure 2.1. The set P in Figure 2.1(a) is convex since every line segment that joins two points in P is within P, such as the line segment ab. The set P in Figure 2.1(b) is not convex since not every line segment that joins two points in P is within P, with an example being the line segment ab as shown in Figure 2.1(b). Having defined the concept of convex set, we can now introduce the notion of convex hull.

Definition 2.2. *(Convex hull). The convex hull of a set P of points in $\mathbb{R}^2$ is the intersection of all convex sets that contain P.*

An example of a convex hull is shown in Figure 2.2. Given a set P of points as shown in Figure 2.2(a), the convex hull of P is shown in Figure 2.2(b). The convex hull of P may be visualized as

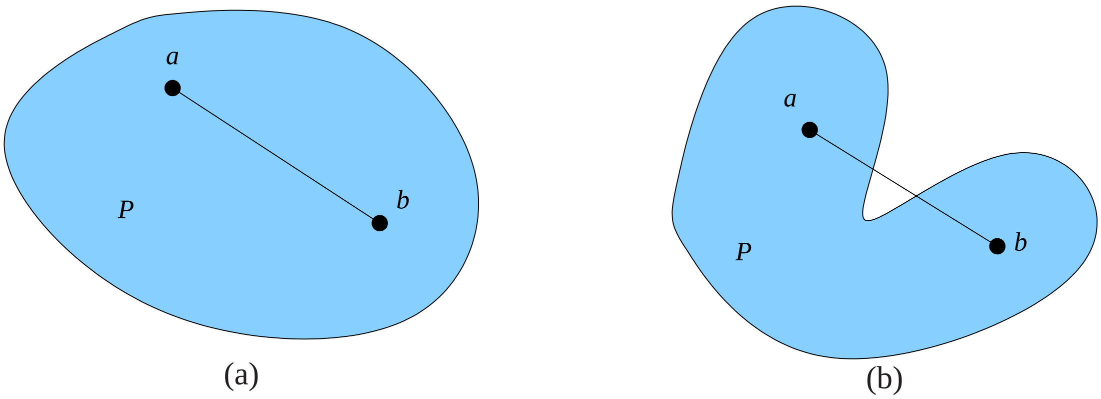


Figure 2.1: Examples of (a) convex and (b) nonconvex sets.

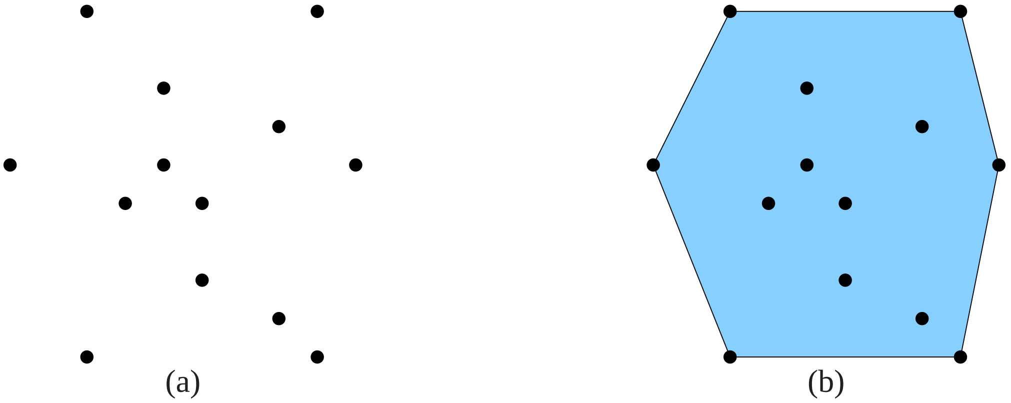


Figure 2.2: Convex hull example. (a) A set P of points, and (b) the convex hull of P.

the shape formed by a rubber band stretched around P. With the definition of convex hull in place, we can now define the concept of triangulation.

Definition 2.3. *(Triangulation). A triangulation of a set P of points in $\mathbb{R}^2$ is a set T of (nondegenerate) triangles such that:*

1. *the union of the vertices of all triangles in T is P;*
2. *the interiors of any two triangles in T are disjoint; and*
3. *the union of the triangles in T is the convex hull of P.*

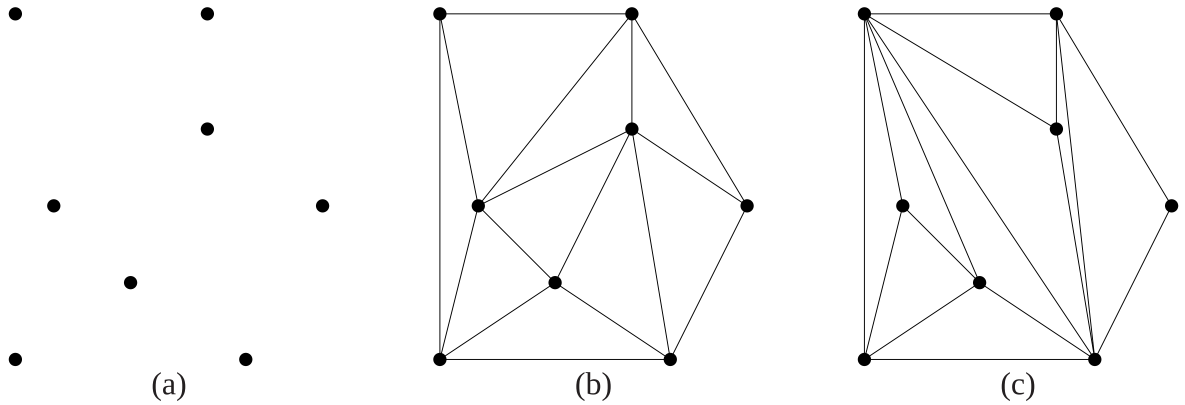


Figure 2.3: Triangulation example. (a) A set *P* of points, (b) a triangulation of *P*, and (c) another triangulation of *P*.

Given a set *P* of points, as shown in Figure 2.3(a), there exist numerous triangulations of *P*, two examples of which are shown in Figures 2.3(b) and (c). We can see that although the triangulations in Figures 2.3(b) and (c) have the same vertices, they differ in how those vertices are connected by edges.

As mentioned in Section 1.2, two of the most commonly employed approaches to choosing the triangulation connectivity for triangle mesh models are Delaunay triangulations [43] and DDTs. We first define the concept of Delaunay triangulation as follows.

Definition 2.4. *(Delaunay triangulation). A Delaunay triangulation of a set P of points in* $\mathbb{R}^2$ *is a triangulation T such that no point in P is inside the circumcircle of any triangle in T.*

Given a set *P* of points as shown in Figure 2.4(a), the Delaunay triangulation of *P* is shown in Figure 2.4(b), with the circumcircle of each triangle shown using a dashed line. As we can see, no vertex of the triangulation falls inside any circumcircle. For a set *P* of points, as long as no four points in *P* are cocircular, the Delaunay triangulation of *P* is uniquely determined, as in the example given in Figure 2.4.

In the case of Delaunay triangulations, the connectivity is determined solely by the set of points being triangulated. For this reason, triangulation-connectivity-selection algorithms employing Delaunay triangulations are typically computationally efficient. The Delaunay triangulation maximizes the

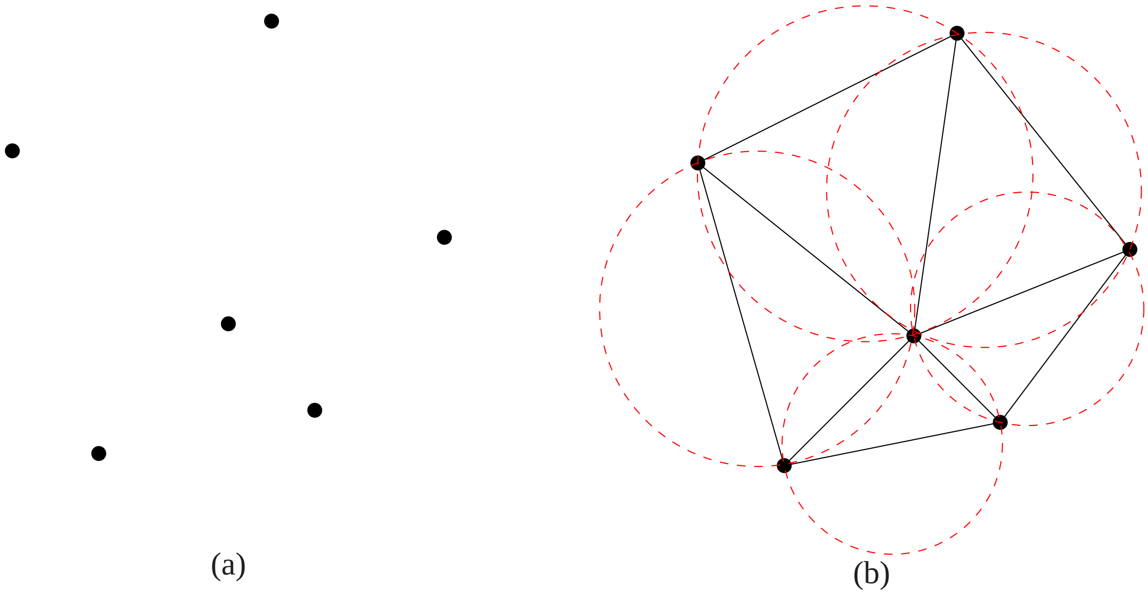


Figure 2.4: Delaunay triangulation example. (a) A set P of points, and (b) the Delaunay triangulation of P.

minimum interior angle of all triangles in the triangulation, thus avoiding sliver triangles to whatever extent possible. This leads to the Delaunay triangulation being favored in approximation applications [44]. DDTs, unlike Delaunay triangulations, allow the triangulation connectivity to be chosen in an arbitrary manner, using information in the dataset from which the points to be triangulated were chosen. Therefore, DDTs offer much more flexibility than Delaunay triangulations.

2.5 Mesh Models of Images

In the context of this thesis, an image is an integer-valued function ϕ defined on the domain $I = [0, W-1] \times [0, H-1]$ and sampled on the truncated two-dimensional integer lattice $\Lambda = \{0, 1, \ldots, W-1\} \times \{0, 1, \ldots, H-1\}$ (i.e., a rectangular grid of width W and height H). A (triangle) mesh model of ϕ consists of:

1. a set $P = \{p_i\}$ of ***sample points***, where $P \subset \Lambda$;

2. a triangulation T of P; and

3. the function values $\{z_i = \phi(p_i)\}$ for $p_i \in P$.

In order to ensure that the triangulation T covers all points in Λ, P must always be chosen to include all of the extreme convex hull points of I (i.e., the four corner points of the image bounding box). As a matter of terminology, the ***size*** and ***sampling density*** of the model are defined as $|P|$ and $|P|/|\Lambda|$, respectively.

The above mesh model is associated with a function $\hat{\phi}$ that approximates ϕ, where $\hat{\phi}$ is determined as follows. First, we construct a continuous piecewise linear function $\tilde{\phi}$ that interpolates ϕ at each point $p_i \in P$. More specifically, for each face f in the triangulation T, $\tilde{\phi}$ is defined to be the unique linear function that interpolates ϕ at the three vertices of f. Since ϕ is integer valued, we wish for its approximation $\hat{\phi}$ to be integer valued as well. Thus, we define the approximation $\hat{\phi}$ in terms of $\tilde{\phi}$ as $\hat{\phi}(p) = \operatorname{round}(\tilde{\phi}(p))$, where round denotes an operator that rounds to the nearest integer.

We provide an example in Figure 2.5 to illustrate the (triangle) mesh modelling of images. Given an image ϕ as shown in Figure 2.5(a) (which can be represented as a surface in Figure 2.5(b)), a triangulation is formed by partitioning the image domain of ϕ into a set of triangles, as illustrated in Figure 2.5(c), with the vertices of the triangulation being the sample points. The resulting triangle mesh model of ϕ is shown in Figure 2.5(d). Then, an approximation of ϕ, known as $\hat{\phi}$, can be generated from the mesh model by standard rasterization techniques [53], with the reconstructed image $\hat{\phi}$ shown in Figure 2.5(e).

In our work, for a given model size (i.e., number of sample points), we want to find a model to minimize ε, the difference between $\hat{\phi}$ and ϕ as measured by the **mean squared error (MSE)**, where

$$\varepsilon = |\Lambda|^{-1} \sum_{p \in \Lambda} \left(\hat{\phi}(p) - \phi(p)\right)^2. \tag{2.3}$$

For convenience, we will express the MSE in terms of the PSNR, which is defined as

$$\mathsf{PSNR} = 20 \log_{10}[(2^{\rho} - 1)/\sqrt{\varepsilon}],$$

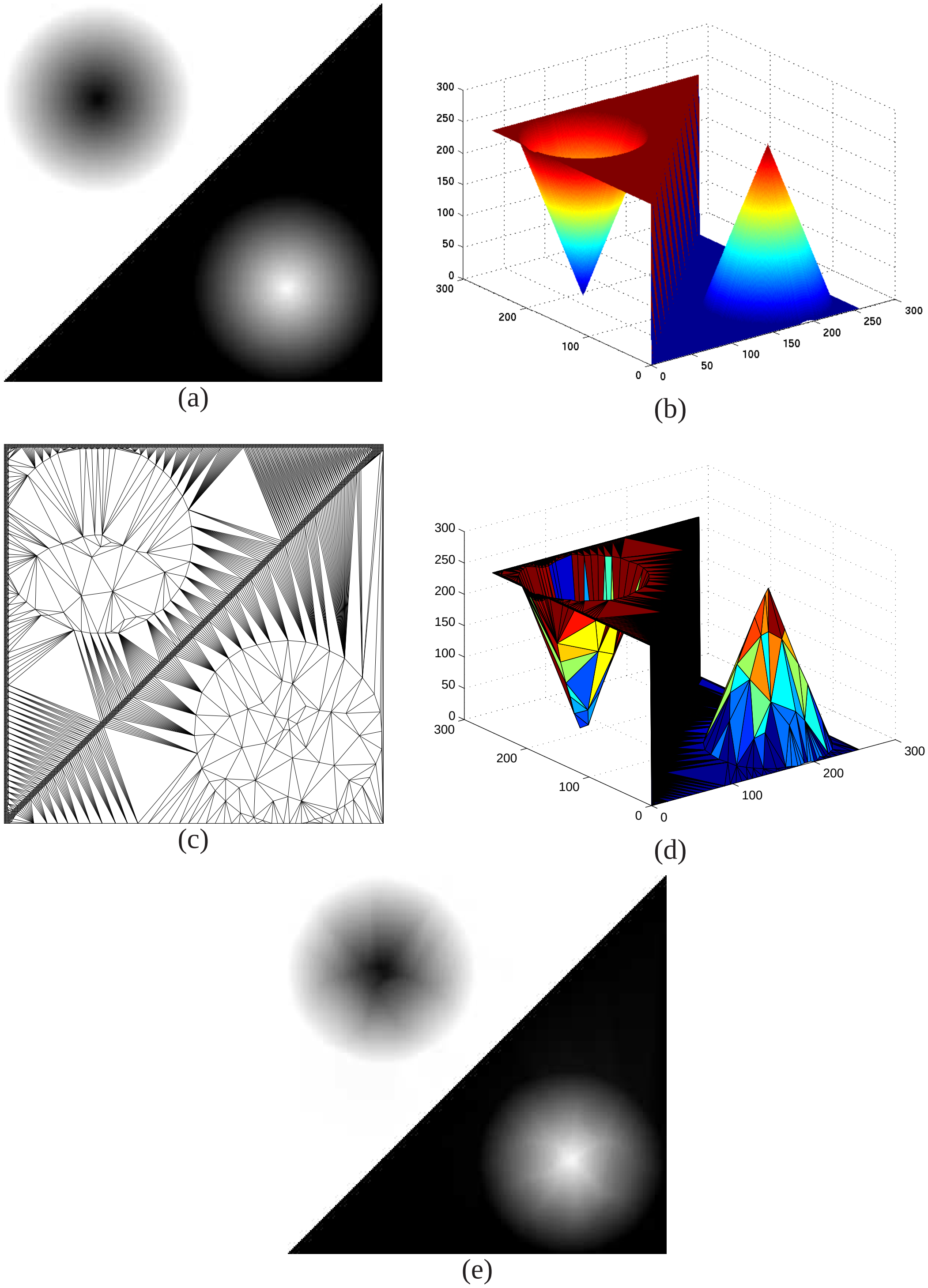


Figure 2.5: Mesh model of an image. (a) Image, (b) image modeled as surface, (c) triangulation of image domain, (d) resulting triangle mesh, and (e) reconstructed image.

where ρ is the number of bits per sample used by the (integer-valued) image ϕ. Finding computationally efficient methods to solve the above problem is extremely challenging, as problems like this are known to be NP-hard [54].

2.6 Grid-Point to Face Mapping

Suppose that we have an image defined on a rectangular grid and a triangulation T superimposed on the grid, let $\Gamma(T)$ denote the set of all integer grid points falling inside or on the boundary of T. For reasons that will become clear later, we need a convenient scheme that can map each point $p \in \Gamma(T)$ to *exactly one* face of the triangulation T. In this thesis, we apply a grid-point to face mapping scheme that is slightly modified from the method proposed in [53]. In particular, a grid point $p \in \Gamma(T)$ is *uniquely* mapped to a face of the triangulation T according to the following rules:

1. If p is strictly inside a face f, map p to f.

2. If p is on an edge e excluding the endpoints of e:

   (a) If e is horizontal, map p to the face below e unless there is no face below, in which case p is mapped to the face above e.

   (b) If e is not horizontal, map p to the face to the left of e unless there is no such face, in which case p is mapped to the face to the right of e.

3. If p is a vertex:

   (a) If p is the right endpoint of a horizontal edge e, map p to the face below e unless there is no face below, in which case p is mapped to the face above e.

   (b) If p is not the right endpoint of any horizontal edge, map p to the face to the left of p unless there is no such face, in which case p is mapped to the face to the right of p.

We provide an example in Figure 2.6 illustrating the mapping rules described above. Given an image defined on a rectangular grid with the extreme convex-hull points of the image domain being $\{v_k\}_{k=0}^{3}$, a triangulation of the points $\{v_k\}_{k=0}^{7}$ is superimposed on the grid, as shown in Figure 2.6(a). In order to facilitate a better understanding of the above mapping rules, the grid points in Figure 2.6(b) are marked with different symbols, with the ones that are mapped to each face sharing the same symbol.

In what follows, we examine Figure 2.6(b) in detail. To begin, let us consider the grid point p_0. The grid point p_0 is strictly inside the face f_3. Thus, according to rule 1, p_0 is mapped to the face f_3. The grid points that fall on an edge or is a vertex are mapped in a more complicated manner. Consider the grid point p_1, which is on the horizontal edge v_1v_2 but not the endpoints of v_1v_2. According to rule 2a, p_1 is mapped to the face f_3 (i.e., the face above the edge v_1v_2) since there is no face below v_1v_2. For the grid point p_2, since it is on the non-horizontal edge v_4v_5, p_2 is mapped to the face f_1 (i.e., the face to the left of v_4v_5) based on rule 2b. Now consider the grid points v_3 and v_4. The grid point v_3 is a vertex and is the right endpoint of the horizontal edge v_7v_3. Therefore, according to rule 3a, v_3 is mapped to the face f_7 (i.e., the face below the edge v_7v_3). The grid point v_4 is a vertex but is not the right endpoint of an edge, so v_4 is mapped to the face f_0 (i.e., the face to the left of v_4) according to rule 3b.

2.7 ED Method

As mentioned earlier, one highly effective method for generating mesh models of images is the ED method [19]. Since our work builds on the ED method, it is helpful to briefly introduce this method here. Given an image ϕ and a desired mesh size N, the ED method constructs a mesh model of ϕ with the set P of sample points, as follows:

1. Sample-point selection. Select P, with $|P| = N$, using Floyd-Steinberg error diffusion [55]. This is done in such a way as to ensure that the points in P are distributed with a density

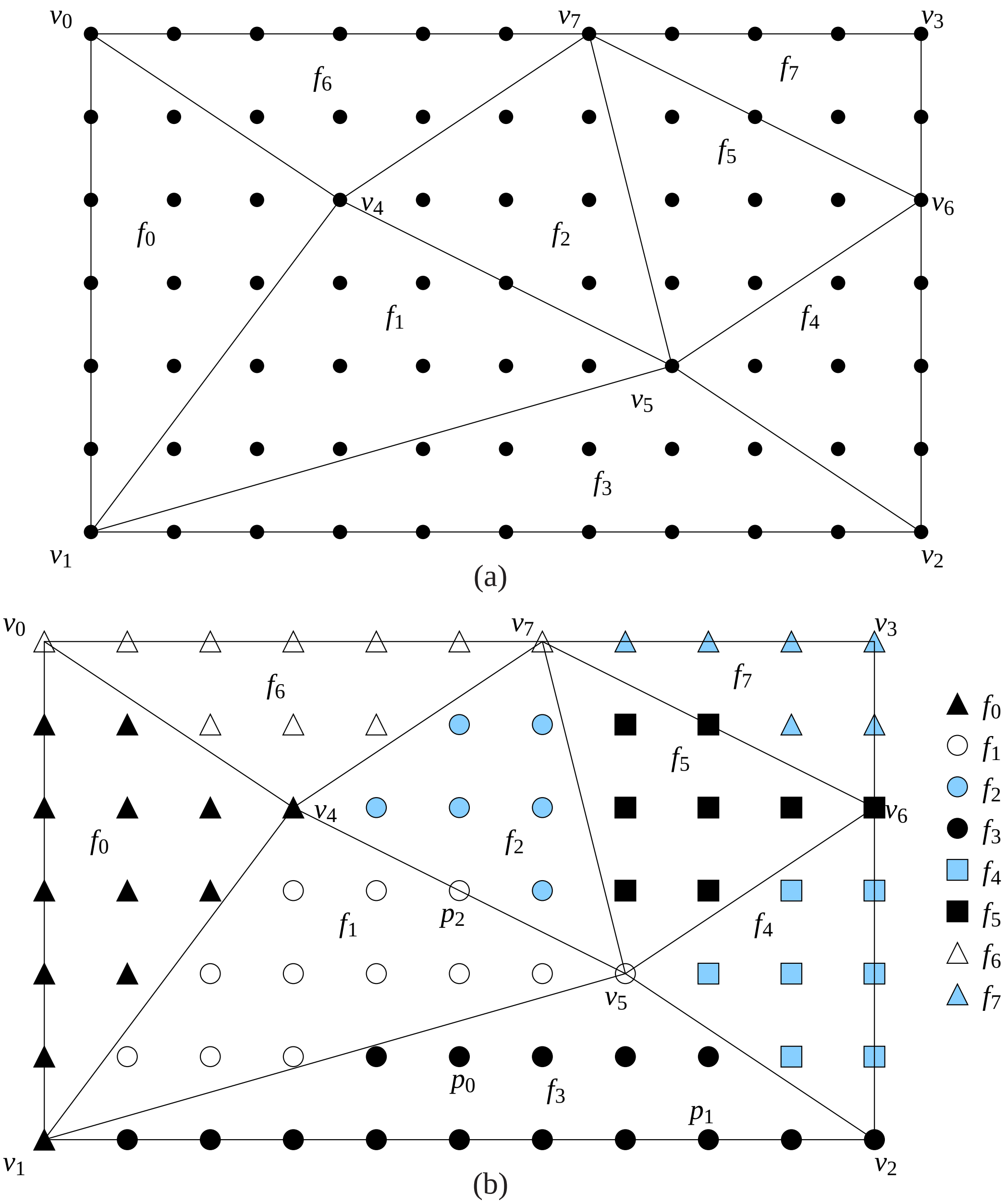


Figure 2.6: An example of grid-point to face mapping. (a) A triangulation on a rectangular grid, and (b) the illustration of how the grid points are uniquely mapped to the faces in the triangulation.

approximately proportional to the MMSODD of ϕ, as given earlier by (2.1).

2. Triangulation. Triangulate P using a Delaunay triangulation.

In step 1, the set P is always chosen to include all extreme convex-hull points of the image domain. This ensures that the triangulation produced in step 2 covers the entire image domain. Since several variants of the ED scheme are presented in [19], it is worth noting, for the sake of completeness, that we consider the variant with the following characteristics herein:

1. a third-order binomial filter is used for smoothing;
2. non-leaky error diffusion is used with a serpentine scan order;
3. the sensitivity parameter γ is chosen as 1; and
4. the error diffusion algorithm is performed iteratively in order to achieve exactly the desired number of sample points.

Since, in our work herein, we require that the approximating function (i.e., $\hat{\phi}$) interpolate the original (i.e., ϕ), we consider only the variant of the ED method that satisfies this interpolating condition. (That is, the variant that employs a least-squares fit is not considered.)

In order to show the sample-point selection strategy (i.e., step 1 above) in the ED method, we provide an illustrative example. Suppose that we have an image ϕ as shown in Figure 2.7(a), where the MMSODD of ϕ is shown in Figure 2.7(b). The sample points that are selected in step 1 above using Floyd-Steinberg error diffusion are shown in Figure 2.7(c), which are distributed with a density approximately proportional to the MMSODD of ϕ. Examining Figure 2.7(c), we can see that the selected sample points mostly concentrate around image edges, representing the shapes and contours of the image ϕ reasonably well.

2.8 Local Optimization Procedure (LOP)

Before proceeding further, it is necessary to interject some additional background related to triangulations. An edge e of a triangulation is said to be ***flippable*** if e has two incident faces (i.e., is not on

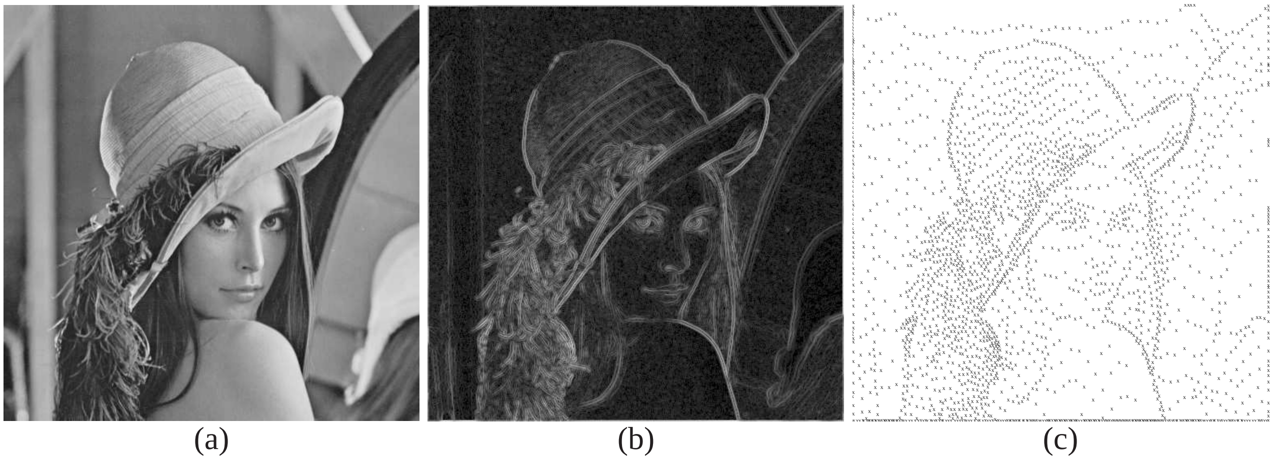


Figure 2.7: (a) An image ϕ, (b) the MMSODD of ϕ , and (c) the sample points selected in step 1.

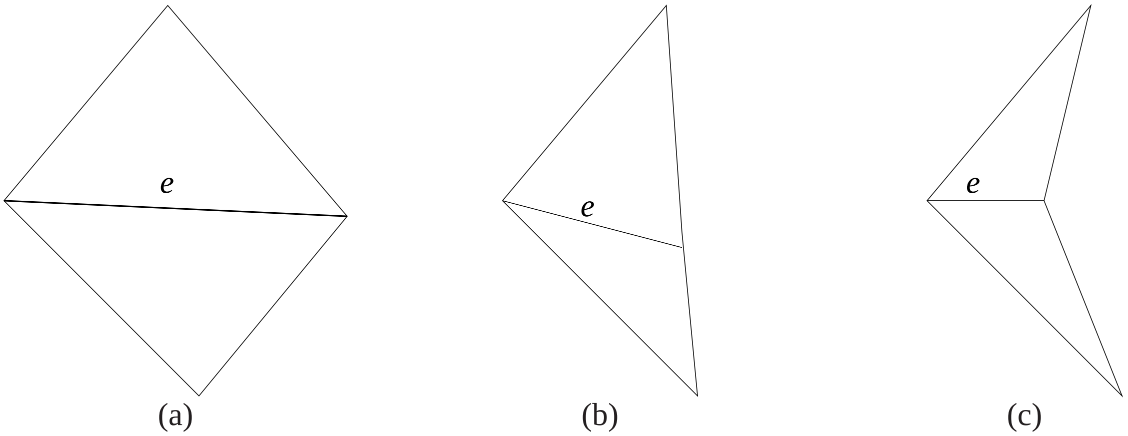


Figure 2.8: Quadrilateral examples. The edge e is the diagonal of a quadrilateral that is (a) strictly convex, (b) convex but not strictly convex, and (c) not convex.

the triangulation boundary) and the union of these two faces is a strictly convex quadrilateral. Figure 2.8 gives examples of flippable and nonflippable edges. In particular, the edge e in Figure 2.8(a) is flippable, while the edge e is not flippable in both Figures 2.8(b) and (c). For a flippable edge e being the diagonal of a strictly convex quadrilateral q that is the union of the two incident faces of e, an ***edge flip*** is an operation that replaces the edge e in the triangulation by the other diagonal e' of q, as shown in Figure 2.9. The fact that every triangulation of a set of points is reachable from every other triangulation of the same set of points via a finite sequence of edge flips [56] motivated Lawson

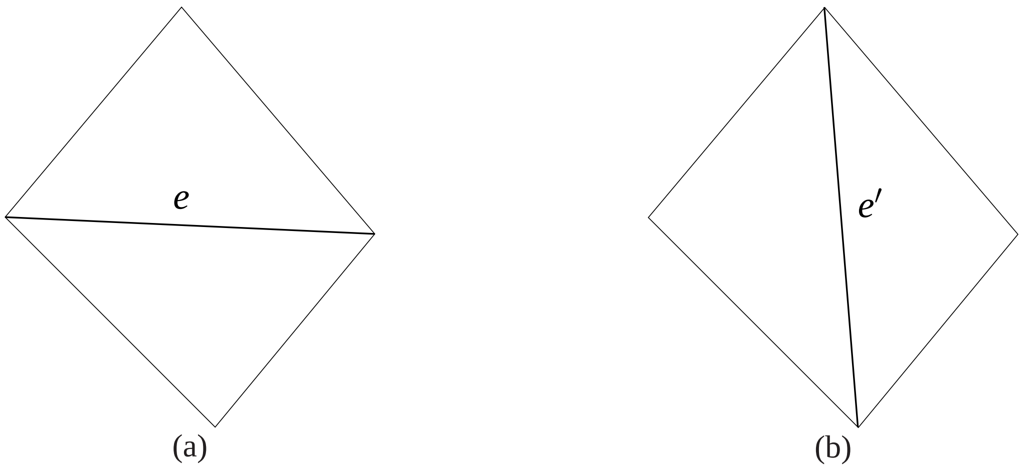


Figure 2.9: An edge flip. (a) Part of the triangulation containing a flippable edge e. (b) The same part of the triangulation after e has been flipped to yield the new edge e'.

to propose the so called LOP [46].

The LOP [46] is an optimization technique, based on edge flips, that is used to select the connectivity of a triangulation so as to be optimal in some sense. In practice, the LOP is frequently used to choose triangulation connectivity in the case of DDTs. As a matter of terminology, a flippable edge e is said to be ***optimal*** if it satisfies some prescribed edge-optimality criterion. In turn, a triangulation T is said to be ***optimal*** if every flippable edge in T is optimal. In order to produce an optimal triangulation, the LOP simply applies edge flips to flippable edges that are not optimal, until the triangulation is optimal (i.e., all flippable edges are optimal).

Cost-Based Criteria. Most frequently, the edge-optimality criterion is specified indirectly through some measure of triangulation cost. Let $\mathrm{triCost}(T)$ denote the cost of the triangulation T. A flippable edge e in the triangulation T is then said to be ***optimal*** if

$$\mathrm{triCost}(T) \leq \mathrm{triCost}(T'), \tag{2.4}$$

where T' is the new triangulation obtained by applying an edge flip to e (in the triangulation T). That is, the flippable edge e is deemed optimal if applying an edge flip to e would not result in a strict decrease in the triangulation cost. In turn, the triangulation cost triCost is specified by defining a cost measure for (all) edges in the triangulation. Let $\text{edgeCost}(T,e)$ denote the cost of the edge e in the triangulation T. Then, triCost is defined as

$$\text{triCost}(T) = \sum_{e \in \mathcal{E}(T)} \text{edgeCost}(T,e), \tag{2.5}$$

where $\mathcal{E}(T)$ denotes the set of edges in T. That is, the cost of a triangulation is simply the sum of its corresponding edge costs. As a matter of terminology, we refer to a triangulation optimality criterion employing (2.4) (where triCost is of the form of (2.5)) as ***cost based***. By far, cost-based criteria are most commonly used in conjunction with the LOP, several examples of which can be found in [23, 57, 25, 26]. A particularly important criterion of this type is **squared error (SE)** [25, 26]. With the SE criterion, the edge e is deemed optimal if applying an edge flip to e would not cause a strict decrease in the MSE as defined by (2.3).

Heuristic-Based Criteria. More recently, the paper [30] introduced a type of triangulation optimality criterion that is not associated with any underlying triangulation cost function (i.e., a function of the form of (2.5)). With this type of criterion, a cost is assigned to each flippable edge. Let $\text{edgeCost}(T,e)$ denote the cost of the edge e in the triangulation T. The flippable edge e is said to be ***optimal*** if

$$\text{edgeCost}(T,e) \leq \text{edgeCost}(T',e'), \tag{2.6}$$

where e' is the new edge produced by applying an edge flip to e and T' is the corresponding new triangulation (with e'). As a matter of terminology, we refer to a triangulation optimality criterion using (2.6) as ***heuristic based***.

Additional Remarks on the LOP. At this point, it is worthwhile to make a few additional remarks about the LOP. The first comment to be made is with respect to algorithm termination. If a cost-based optimality criterion is employed, the LOP must terminate after a finite number of steps (assuming the algorithm is implemented in a numerically robust manner). This is an indirect consequence of the fact that the LOP only flips an edge if doing so would result in a strict decrease in the triangulation cost. In contrast, if a heuristic-based optimality criterion is used (regardless of whether the implementation is numerically robust), the LOP can potentially become trapped in a cycle, repeating the same sequence of edge flips indefinitely. This is due to the fact that, in the absence of a well-defined triangulation cost function, it is possible to make inconsistent decisions about the optimality of an edge. Such inconsistent decisions can result in cycles. From a practical standpoint, this potential cycling issue does not pose any significant problems for two reasons. First, when performing the LOP, it is easy to avoid being trapped in a cycle by simply tracking how many times each edge is tested for optimality and if the count for an edge exceeds a particular threshold some special action can be taken, such as ignoring the edge for the remainder of the LOP or terminating the LOP early. Second, the more effective heuristic-based criteria only rarely result in cycles. Therefore, breaking cycles when they do occur has little impact on the result produced by the LOP. In the implementation employed in our work, in the case of heuristic-based criteria, we limit the number of times an edge may be tested for optimality to 15. If this count is exceeded, the edge in question is simply ignored for the remainder of the LOP.

The second remark to make about the LOP concerns the optimal triangulation that it produces. For any optimality criterion of practical interest (other than the Delaunay criterion [43, 45]), the optimal solution produced by the LOP is almost never uniquely determined. The nonuniqueness of the solution is important because it implies that some optimal solutions may be (and, in practice, are) much better than others. The optimum produced will typically depend (often very heavily) on the initial triangulation to which the LOP is applied.

2.9 Look-Ahead LOP (LLOP)

Suppose that the LOP is used in conjunction with a cost-based optimality criterion. In this case, if a triangulation T is optimal, then no single edge flip can result in a new triangulation with strictly lower cost than T. If, however, more than one edge flip is allowed, it can no longer be guaranteed that the triangulation cost will not strictly decrease. In this sense, the LOP only guarantees a locally (but not necessarily globally) optimal triangulation. Since some local minima will, in practice, have a much lower cost than others, it would be advantageous to have some means to reduce the likelihood of converging to a poor local minimum. This observation motivated Yu et al. to propose the so called LLOP [29].

The LLOP is similar to the LOP in that the LLOP applies edge-flip-based transformations to a triangulation until the triangulation is optimal. The LLOP, however, differs from the LOP in two key respects. The first difference is that, instead of only allowing the triangulation to be transformed by a single edge flip in each step, the triangulation can be transformed by: 1) a single edge flip; or 2) a sequence of two edge flips, where the two edges involved share a common face. The second difference is that the definition of triangulation optimality is changed to the following: A triangulation T is said to be ***optimal*** if the application of a single transformation of one of the two above types cannot produce a new triangulation whose cost is strictly less than that of T.

For the benefit of the reader, we provide an example illustrating the two types of triangulation transformation allowed in the LLOP. Consider part of a triangulation as shown in Figures 2.10(a) and 2.11(a). As described above, this part of the triangulation can be transformed by: 1) a single edge flip; or 2) a sequence of two edge flips, where the two edges involved share a common face. To begin, let us consider an example illustrating the first type of triangulation transformation in the LLOP. Given part of a triangulation as shown in 2.10(a), Figure 2.10(b) shows the resulting triangulation by applying a single edge flip to the edge e in Figure 2.10(a). Then, let us consider an example of the second type of triangulation transformation in the LLOP. Given part of a triangulation as shown in 2.11(a), this part of the triangulation can be transformed into one of the four triangulations shown in Figures 2.11(c)–(f), via a sequence of two edge flips.

By being allowed to apply sequences of two edge flips (instead of just individual edge flips), the LLOP is able to reduce the likelihood of converging to a very poor local minimum. In effect, when trying to minimize the triangulation cost, the LLOP considers the effect of not just single edge flips

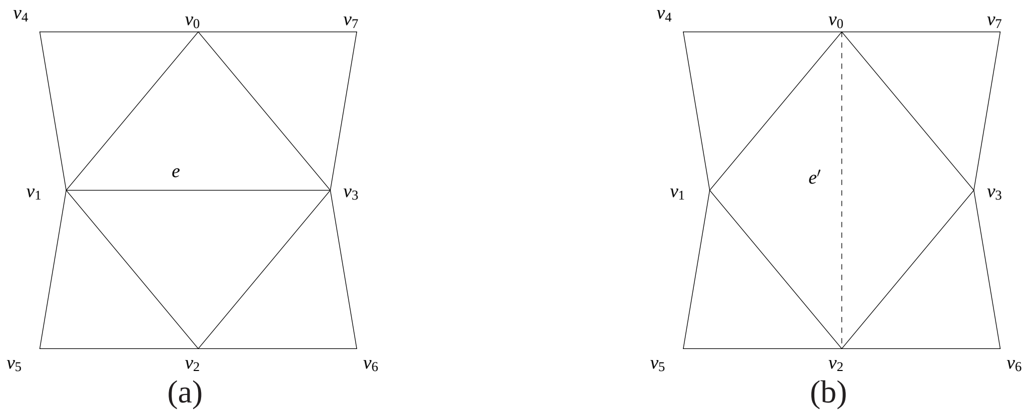
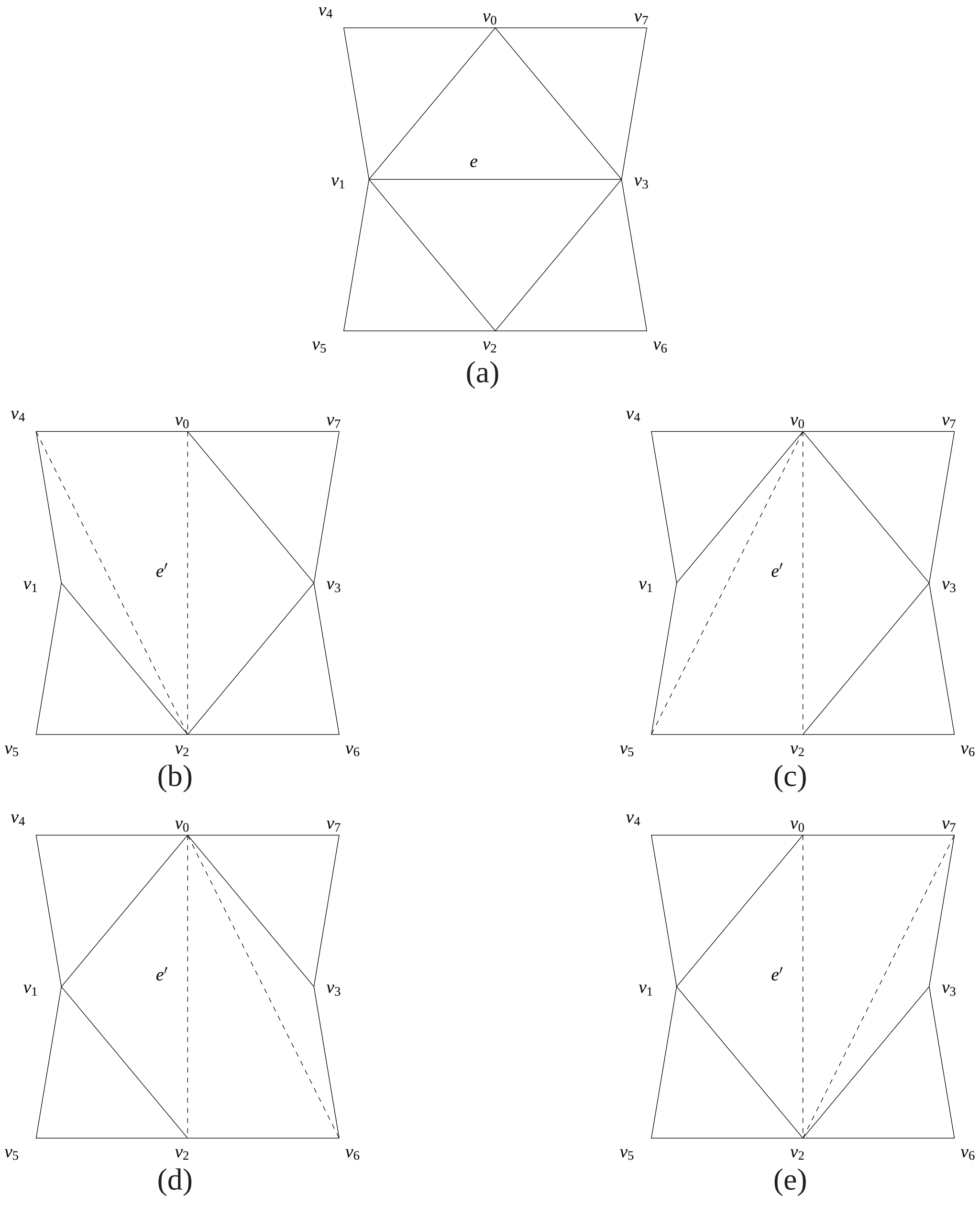


Figure 2.10: An example of the first type of triangulation transformation in the LLOP. (a) Part of the triangulation containing a flippable edge e. (b) The same part of the triangulation after applying a single edge flip to e.

(like the LOP) but also sequences of two edge flips. In practice, the LLOP usually produces a better local optimum (i.e., a triangulation with lower cost) than the LOP. The downside of the LLOP is that it typically requires more computation time and can be quite difficult to implement in a numerically robust manner. Since the LLOP fundamentally relies on the existence of a triangulation cost function, the LLOP can only be used in conjunction with optimality criteria that are cost based. In other words, the LLOP cannot be used with heuristic-based optimality criteria.



(a)

(b)

(c)

(d)

(e)

Figure 2.11: An example of the second type of triangulation transformation in the LLOP. (a) Part of the triangulation containing a flippable edge *e* before transformation. (b)–(e) Four possible resulted cases of the same part of the triangulation after applying a sequence of two edge flips.

Chapter 3

Proposed Approach

3.1 Overview

In this chapter, we propose a computational framework modified from the ED method and derive two proposed methods from our framework. We begin by introducing the computational framework with several free parameters. Then, we present various choices for each of the free parameters and advocate two particular choices of those parameters, leading to our two proposed methods. When discussing our framework, we also propose two optimality criteria for the LOP that perform extremely well. After that, we evaluate our proposed methods in terms of mesh quality and computational complexity. Lastly, for the completeness of the thesis, we briefly discuss some extra work we have done during the development of our framework.

3.2 Computational Framework for Mesh Generation

Having introduced the necessary background in Chapter 2, we now turn our attention to introducing the two mesh-generation methods proposed in this thesis. As explained earlier, the ED method chooses triangulation connectivity using a Delaunay triangulation. Experimentally, however, we have observed that selecting the connectivity in this way results in a mesh in which triangulation edges often crosscut image edges (i.e., discontinuities in the image), leading to a degradation in approximation quality. This motivated us to consider choosing triangulation connectivity in a more flexible manner, using a DDT instead of a Delaunay triangulation.

In what follows, we will first introduce our general computational framework for mesh generation, which has several free parameters. Then, by advocating two particular choices for these parameters, we will arrive at the two specific mesh-generation methods proposed herein, namely MED1 and MED2. Since it is helpful for the reader to see how we arrived at these choices, we provide significant detail in this regard, including some experimental results.

Given an image ϕ and a desired mesh size N as input, our general computational framework for mesh generation produces a mesh model of ϕ having the set P of sample points, with $|P| = N$, and the associated triangulation T. To accomplish this objective, our framework performs the following (in order):

1. Sample-point selection. Select P using the same sample-point selection strategy in step 1 of the ED method (as introduced earlier in Section 2.7 on page 17).
2. Initial mesh construction. For each point $p \in P$ using the order specified by insOrder, where insOrder is a free parameter of the framework:
   (a) Insert p in the triangulation T. This is accomplished by deleting any faces containing p and retriangulating the resulting hole. This point-insertion process is illustrated in Figures 3.1 and 3.2.
   (b) Adjust the connectivity of T by applying the LOP (as described in Section 2.8 on page 19) with the triangulation optimality criterion chosen as insOptCriterion, where insOptCriterion is a free parameter of our framework.
3. Final connectivity adjustment. Adjust the connectivity of T by applying either the LOP or LLOP, as specified by the parameter fcaMethod, with the optimality criterion chosen as SE

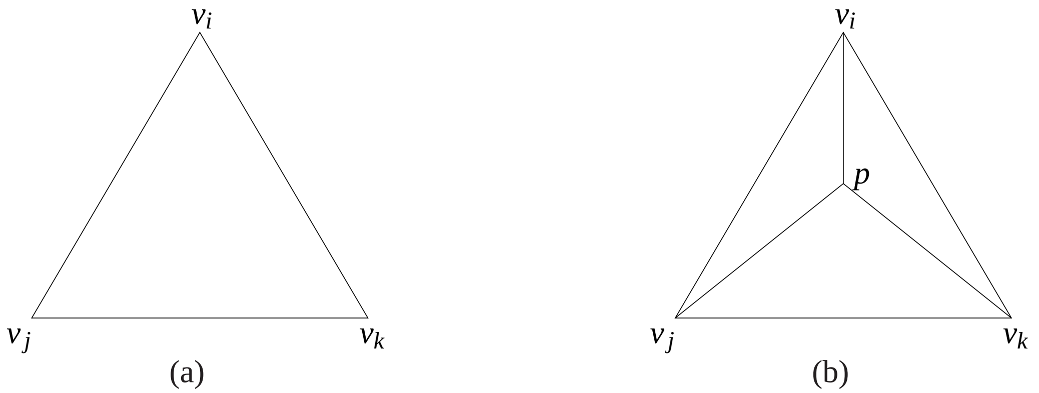


Figure 3.1: Insertion of a point strictly inside a face. (a) Part of the triangulation showing a triangle $v_iv_jv_k$. (b) The same part of the triangulation with a new vertex p inserted inside the triangle $v_iv_jv_k$.

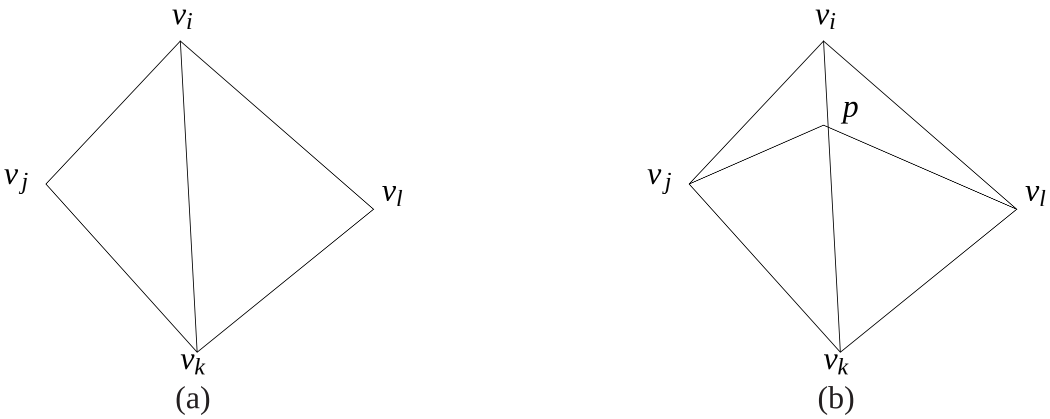


Figure 3.2: Insertion of a point on an edge. (a) Part of the triangulation showing two incident faces $v_iv_jv_k$ and $v_iv_kv_l$. (b) The same part of the triangulation with a new vertex p inserted on the edge v_iv_k.

(i.e., squared error). If fcaMethod is LOP, the LOP is employed in this step; otherwise (i.e., if fcaMethod is LLOP), the LLOP is used.

In step 2b of the above framework, the choice of the triangulation optimality criterion insOptCriterion is critical, as different choices of insOptCriterion will typically lead to vastly differing meshes. One of the optimality criteria considered in our work is the SE criterion introduced in Section 2.8. We also considered numerous other criteria, which we will introduce shortly. Before proceeding further, however, there is a very important comment that we must make regarding our above framework. Since our objective is to produce a mesh that minimizes the MSE (as given by (2.3)), this suggests the "obvious" solution of choosing the optimality criterion insOptCriterion as SE and simply skipping final connectivity adjustment (i.e., step 3) altogether (since final connectivity adjustment would not change anything if insOptCriterion were chosen as SE). In other words, the obvious solution would be to simply optimize for squared error using the LOP after the insertion of each point in step 2. As it turns out, this obvious solution performs extremely poorly. This poor performance is due to an interplay between point insertion and the SE criterion in step 2b, which leads to triangulations with many poorly-chosen sliver (i.e., long thin) triangles, severely degrading approximation quality. In effect, this interplay causes the mesh-generation optimization process to converge to an extremely poor local optimum. To combat this problem, our framework allows the parameter insOptCriterion to be chosen differently from SE, and then adds a final connectivity-adjustment step employing the SE criterion in order to reduce the squared error for the final mesh.

Insertion Order. Recall that step 2 of our framework (i.e., initial mesh construction) utilizes the parameter insOrder, which specifies the order in which points are to be inserted in the triangulation. In our work, we considered numerous possible choices for the insertion order insOrder, including:

1. randomized order: the extreme convex-hull points followed by the remaining points in randomized order;

2. xy-lexicographic order: the extreme convex-hull points followed by the remaining points in xy-lexicographic order;

3. farthest-point first order: the extreme convex-hull points followed by the remaining points prioritized such that the point most distant from the vertices in the triangulation is inserted first; and

4. closest-point first order: the extreme convex-hull points followed by the remaining points prioritized such that the point nearest another vertex in the triangulation is inserted first.

Detailed experiments showed randomized order (i.e., item 1 above) to be most effective. In particular, we found that, relative to randomized order, no one of the other insertion orders considered was able to consistently produce higher quality meshes at lower or comparable computational cost. Consequently, we advocate that insOrder always be chosen as randomized order, and we assume that this choice is always made for the remainder of this thesis.

Optimality Criteria. Recall that step 2b of our framework (i.e., connectivity adjustment after point insertion) utilizes the parameter insOptCriterion, which determines the particular triangulation optimality criterion used for connectivity adjustment. In our work, we considered the following twelve possibilities for the choice of the optimality criterion insOptCriterion:

1. **squared error (SE)** [25, Equation 1] and [26, Section 2];
2. (preferred-direction) Delaunay [45, Section 2] and [46, Section 11.2];
3. **angle between normals (ABN)** [23, Equation 3];
4. **jump in normal derivatives (JND)** [23, Section 3.1];
5. **deviations from linear polynomials (DLP)** [23, Section 3.1];
6. **distances from planes (DP)** [23, Section 3.1];
7. **absolute mean curvature (AMC)** [57, Section 2.2];
8. **Garland-Heckbert hybrid (GHH)** [18, Algorithm IV and Section 4.5.1] and [30, Section III.B];
9. **shape-quality-weighted SE (SQSE)** [30, Section III.B];
10. **JND-weighted SE (JNDSE)** [30, Section III.B];
11. **edge-length-weighted SE (ELSE)** [proposed herein]; and
12. **minimum-angle-weighted SE (MASE)** [proposed herein].

The first ten of the above criteria are well known criteria taken from the literature, while the remaining two (namely, ELSE and MASE) are newly proposed in this thesis. In the interest of brevity, we will only present herein the formal mathematical definitions of the two new criteria. The definitions of the other optimality criteria can be found in the references provided above. Of the old criteria (i.e., the first ten) the SE, Delaunay, ABN, JND, DLP, DP, and AMC criteria are all cost based (i.e., employ (2.4)), the SQSE and JNDSE criteria are heuristic based (i.e., employ (2.6)), and the GHH criterion is a hybrid of two cost-based criteria.

Before formally defining the ELSE and MASE criteria, we must first introduce some additional notation. For a triangulation T, let $\Gamma(T)$ denote the set of all integer lattice points falling inside or on the boundary of T. For a given triangulation T, let face_T denote a function that maps each point $p \in \Gamma(T)$ to *exactly one* face in T, where this function is defined using the grid-point to face mapping rules introduced in Section 2.6 (on page 16). The set of all points $p \in \Gamma(T)$ satisfying $\mathrm{face}_T(p) = f$ is denoted $\mathrm{points}_T(f)$. With this notation in place, we can now proceed to present the ELSE and MASE criteria.

The ELSE and MASE criteria are both heuristic based (i.e., employ (2.6)). Therefore, each of these criteria is completely specified in terms of an edge-cost function. For a flippable edge e in the triangulation T, the edge-cost functions for the ELSE and MASE criteria are given, respectively, by

$$\mathrm{edgeCost}_{\mathsf{ELSE}}(T,e) = \|e\| \left[\beta(T,f_i) + \beta(T,f_j)\right] \quad \text{and} \tag{3.1a}$$

$$\mathrm{edgeCost}_{\mathsf{MASE}}(T,e) = \frac{\beta(T,f_i) + \beta(T,f_j)}{\min\left\{\theta(f_i),\theta(f_j)\right\}}, \tag{3.1b}$$

where

$$\beta(T,f) = \sum_{p \in \mathrm{points}_T(f)} \left(\hat{\phi}(p) - \phi(p)\right)^2,$$

f_i and f_j denote the two faces incident to e, $\theta(f)$ denotes the minimum interior angle of the face f, and points_T is as defined earlier.

Final Connectivity Adjustment. In step 3 of our framework, the fcaMethod parameter is used to select whether the LOP or LLOP is used for final connectivity adjustment. Having the ability to choose between the LOP and LLOP provides us with more flexibility to trade off between mesh quality and computational cost. In case the reader might be wondering why we did not allow similar flexibility to choose between the LOP and LLOP for connectivity adjustment after point insertion (i.e., in step 2b), we explain our rationale for this decision in what follows. The overriding reason for this decision was that, as we shall see later, all of the most effective triangulation optimality criteria during point insertion (i.e., in step 2b) are heuristic based, and such criteria cannot be used with the LLOP. Consequently, allowing the use of the LLOP during point insertion would not facilitate the development of a better mesh-generation method. To a much lesser extent, our decision was also influenced by computational cost considerations. In particular, much more time is typically spent performing connectivity adjustment in step 2b (in total) than in step 3. Thus, the increase in computational cost resulting from replacing the LOP with the LLOP in step 2b is much higher than that of replacing the LOP with the LLOP in step 3. Due to these (as well as other) factors, our framework only accommodates the use of the LOP in step 2b.

3.3 Test Data

Shortly, we will have the need to present some experimental results obtained with various test images. So, before proceeding further, a brief digression is in order to introduce the test images that we employed. In our work, we have used 40 images, taken mostly from standard test sets such as [58], [59], and [60]. For the most part, the results that we present herein focus on the representative subset of these images listed in Table 3.1. This particular subset was chosen to contain a variety of image types (i.e., photographic, medical, and computer-generated imagery).

Table 3.1: Test images

Image	Size, Bits/Sample	Description
bull	1024×768, 8	cartoon animal
cr	1744×2048, 10	x-ray [58]
lena	512×512, 8	woman [59]

3.4 Selection of Free Parameters

As seen earlier, our computational framework for mesh generation has three free parameters, namely, 1) the insertion order insOrder, 2) the triangulation optimality criterion insOptCriterion, and 3) the method fcaMethod used for final connectivity adjustment. For the reasons presented earlier, we advocate choosing insOrder as randomized order. In what follows, we study the effects of making various choices for the two remaining parameters (namely, insOptCriterion and fcaMethod). Based on this analysis, we ultimately recommend two particular choices for these parameters, leading to our two proposed mesh-generation methods.

Triangulation Optimality Criterion During Point Insertion. To begin, we study how the choice of triangulation optimality criterion insOptCriterion in step 2b of our framework (on page 28) affects mesh quality. Since the best choice of optimality criterion might possibly be dependent on whether the final-connectivity-adjustment method fcaMethod is chosen as LOP or LLOP, we treat these two cases separately. For fcaMethod being chosen as each of LOP and LLOP, we proceeded as follows. For each of the 40 images in our test set and five sampling densities per image (for a total of $40 \cdot 5 = 200$ test cases), we generated a mesh using each of the twelve choices for insOptCriterion under consideration, and measured the resulting approximation error in terms of PSNR. In each of the test cases, the results obtained with the twelve methods were ranked from 1 (best) to 12 (worst). Then, the average and standard deviation of these ranks were computed across each sampling density as well as overall. These ranking results are given in Tables 3.2(b) and 3.3(b) for the cases of fcaMethod being chosen as LOP and LLOP, respectively. Individual results for three specific images (namely, the ones listed in Table 3.1) are provided in Tables 3.2(a) and 3.3(a) for fcaMethod being chosen as LOP and LLOP, respectively. In each of these tables, the best result in each row is shown in bold font.

First, let us examine the results for the case that fcaMethod is chosen as LOP. From the ranking results in Table 3.2(b), we can make several observations: 1) the ELSE criterion is the clear winner (with an overall rank of 1.16), followed by the MASE and JNDSE criteria (with overall ranks of 2.50 and 2.85, respectively); 2) the MASE criterion yields better results than the JNDSE criterion, except at high sampling densities where the two criteria are comparable; and 3) the worst performers are the SE and DP criteria (with overall ranks of 10.88 and 11.74, respectively). Observation 3 supports our earlier claim that the SE criterion leads to extremely poor results (when used during point insertion). To add to observation 1, it is worth noting that a more detailed examination of the results shows that the ELSE criterion performs best and second best in 187/200 (94%) and 4/200 (2%) of the test cases, respectively. This observation is in agreement with the fact that the standard deviations for the rankings for the ELSE criterion are quite small (e.g., 0.89 or less). For that matter, most of the standard deviations in the table are relatively small, indicating that the actual ranking results tend to be reasonably close to the average rank. The results for the individual test cases, shown in Table 3.2(a), are consistent with the ranking results. For example, the ELSE criterion is the best, outperforming the second and third best criteria, MASE and JNDSE, in all 15 test cases by 0.01 to 1.77 dB and 0.04 to 4.15 dB, respectively. Moreover, the MASE criterion outperforms the JNDSE criterion in 12/15 of the test cases by 0.01 to 2.38 dB. In the preceding results, PSNR was found to correlate reasonably well with subjective quality. It is worthwhile to note that the two best performing criteria ELSE and MASE are newly proposed herein. This shows that our ELSE and MASE criteria, especially the former, make an important contribution beyond well-known criteria from the existing literature.

Now, let us consider the results for the case that fcaMethod is chosen as LLOP. As we will see momentarily, the trends in this case are, for the most part, similar to those for the case just studied above. Examining Table 3.3(b), we observe that: 1) the ELSE criteria is the clear winner (with an overall rank of 1.44) followed by the MASE and JNDSE criteria (with overall ranks of 2.41 and 3.46, respectively); and 2) the SE and DP criteria are the worst performers (with overall ranks of 11.31 and 11.47, respectively). To add to observation 1, a more detailed analysis of the results shows the

Table 3.2: Comparison of the mesh quality obtained with the various choices of triangulation optimality criterion insOptCriterion in the case that fcaMethod is LOP. (a) PSNRs for three specific images. (b) Rankings averaged across 40 images.

(a)

Image	Samp. Density (%)	PSNR (dB)											
		SE	Del.	ABN	JND	DLP	DP	AMC	GHH	SQSE	JNDSE	MASE	ELSE
bull	0.5	24.73	31.44	28.44	30.49	29.72	25.96	31.42	30.22	31.59	31.37	33.75	**35.52**
	1.0	26.89	38.85	31.91	38.34	33.42	30.73	39.00	37.82	38.69	38.78	39.43	**39.99**
	2.0	30.53	42.12	34.87	42.26	39.52	25.93	41.94	41.36	42.36	42.36	42.49	**42.72**
	3.0	31.93	43.42	36.24	43.36	39.35	29.91	43.08	43.41	43.61	43.66	43.83	**43.97**
	4.0	31.23	44.34	34.97	44.17	41.40	31.92	44.07	44.23	44.49	44.53	44.62	**44.72**
cr	0.5	31.19	34.40	30.38	34.45	32.42	30.23	34.22	34.30	34.81	34.84	34.92	**35.11**
	1.0	32.41	36.33	33.01	36.35	34.42	31.16	36.37	36.48	37.13	37.16	37.22	**37.31**
	2.0	33.33	38.68	34.34	38.36	36.33	32.52	38.24	38.75	38.95	39.01	39.00	**39.10**
	3.0	34.12	39.57	34.95	39.32	36.96	33.78	39.17	39.62	39.76	39.82	39.80	**39.87**
	4.0	35.63	40.10	39.29	39.89	37.56	33.54	39.70	40.19	40.31	40.36	40.33	**40.42**
lena	0.5	17.61	21.17	19.22	20.55	19.61	18.07	20.51	21.20	21.75	21.82	21.83	**21.96**
	1.0	21.50	25.21	20.69	24.91	21.86	19.91	24.58	25.30	25.89	25.92	25.94	**26.13**
	2.0	20.72	29.48	24.36	29.09	26.25	21.04	27.67	29.26	29.91	29.99	30.06	**30.14**
	3.0	23.43	31.26	24.62	30.99	27.22	22.34	30.15	31.21	31.58	31.62	31.71	**31.72**
	4.0	23.67	32.39	26.30	32.17	29.13	24.06	31.45	32.47	32.78	32.84	32.87	**32.88**

(b)

Samp. Density (%)	Mean Rank[a]											
	SE	Del.	ABN	JND	DLP	DP	AMC	GHH	SQSE	JNDSE	MASE	ELSE
0.5	10.20 (1.42)	5.83 (1.30)	9.78 (1.47)	8.10 (1.67)	8.90 (1.62)	11.40 (1.22)	7.30 (1.68)	5.13 (1.73)	3.85 (1.22)	3.43 (1.66)	2.83 (1.53)	**1.28** (0.89)
1.0	10.95 (0.80)	6.05 (1.18)	10.03 (0.57)	7.60 (0.80)	9.10 (0.49)	11.75 (0.49)	6.95 (1.26)	5.35 (0.79)	3.80 (0.64)	2.98 (0.79)	2.28 (0.63)	**1.18** (0.67)
2.0	10.98 (0.47)	5.80 (0.40)	10.00 (0.50)	7.08 (0.52)	9.10 (0.37)	11.90 (0.30)	7.85 (0.42)	5.30 (0.64)	3.75 (0.54)	2.75 (0.66)	2.43 (0.70)	**1.08** (0.35)
3.0	11.18 (0.44)	5.90 (0.37)	10.00 (0.22)	6.95 (0.22)	9.03 (0.16)	11.80 (0.40)	8.00 (0.00)	5.13 (0.46)	3.88 (0.40)	2.58 (0.67)	2.43 (0.54)	**1.15** (0.69)
4.0	11.10 (0.44)	5.88 (0.33)	10.03 (0.27)	7.03 (0.16)	9.03 (0.16)	11.85 (0.36)	7.98 (0.16)	5.10 (0.37)	3.85 (0.53)	2.53 (0.55)	2.55 (0.59)	**1.10** (0.62)
All	10.88 (0.88)	5.89 (0.84)	9.97 (0.76)	7.35 (0.97)	9.03 (0.79)	11.74 (0.67)	7.62 (1.05)	5.20 (0.94)	3.83 (0.72)	2.85 (1.01)	2.50 (0.90)	**1.16** (0.67)

[a] The standard deviation is given in parentheses.

ELSE criterion to perform best and second best 166/200 (83%) and 12/200 (6%) of the test cases, respectively. This observation is in agreement with the fact that the standard deviations for the rankings for the ELSE criterion are quite small (e.g., 1.19 in the overall case). For that matter, most of the standard deviations in the table are relatively small, indicating that the actual ranking results tend to be reasonably close to the average rank. Compared to the case when fcaMethod is chosen as LOP, we observe that the MASE criterion outperforms the JNDSE criterion even more consistently (i.e., the two criteria differ more in terms of their overall rankings). The results for individual test cases shown in Table 3.3(a) are consistent with the preceding ranking results. For example, the ELSE criterion is the best, outperforming the second and third best criteria, MASE and JNDSE, in 13/15 of the test cases by 0.01 to 2.22 dB and 0.01 to 2.61 dB, respectively, and the MASE criterion outperforms the JNDSE criterion in 12/15 of the test cases by 0.01 to 0.84 dB. Again, in the preceding results, PSNR was found to correlate reasonably well with subjective quality.

As the above experimental results demonstrate, regardless of whether the final-connectivity-adjustment method fcaMethod is chosen as LOP or LLOP, the best performance in terms of approximation quality is obtained by choosing the triangulation optimality criterion insOptCriterion as ELSE. Therefore, we advocate this particular choice for insOptCriterion in our framework.

In the experimental results above, we saw that, regardless of whether the final-connectivity-adjustment method fcaMethod is chosen as LOP or LLOP, selecting the triangulation optimality criterion insOptCriterion as SE leads to meshes of extremely poor quality. Earlier, we indicated that this behavior is due to an interplay between point insertion and the SE criterion, which leads to triangulations with many poorly-chosen sliver triangles. To illustrate this phenomenon, we present two examples, one for the parameter fcaMethod being chosen as each of LOP and LLOP. For consistency, the examples are taken from the results presented earlier in Tables 3.2 and 3.3, and correspond to the lena image at a sampling density of 2%. For the parameter fcaMethod being chosen as each of LOP and LLOP, the results obtained are shown in Figures 3.3 and 3.4, respectively. Each figure shows part of the image approximation and the corresponding image-domain triangulation obtained when insOptCriterion is chosen as SE. For comparison purposes, the result obtained with the ELSE criterion (which performs

Table 3.3: Comparison of the mesh quality obtained with the various choices of triangulation optimality criterion insOptCriterion in the case that fcaMethod is LLOP. (a) PSNRs for three specific images. (b) Rankings averaged across 40 images.

(a)

Image	Samp. Density (%)	PSNR (dB)											
		SE	Del.	ABN	JND	DLP	DP	AMC	GHH	SQSE	JNDSE	MASE	ELSE
bull	0.5	26.68	34.31	31.70	35.24	31.83	29.98	34.72	33.10	33.75	33.88	34.72	**36.20**
	1.0	30.67	40.41	36.93	40.12	37.30	35.76	40.30	40.01	40.15	40.17	40.56	**42.78**
	2.0	33.73	43.27	39.05	43.37	41.25	31.54	43.39	42.69	43.23	43.23	43.38	**43.51**
	3.0	34.91	44.43	39.85	44.43	42.30	34.84	44.51	44.32	44.39	44.40	44.52	**44.58**
	4.0	33.76	45.20	40.25	45.15	44.34	36.72	45.22	45.05	45.19	45.21	45.29	**45.33**
cr	0.5	32.73	35.88	32.68	35.65	34.05	32.50	35.65	35.76	35.95	35.91	35.92	**36.04**
	1.0	34.13	37.71	35.10	37.18	35.92	33.19	37.55	37.72	37.77	37.78	37.78	**37.82**
	2.0	34.83	39.36	36.51	39.19	37.92	34.76	39.19	39.35	39.40	39.41	39.42	**39.43**
	3.0	35.68	40.14	37.16	40.01	38.59	35.99	39.99	40.12	40.17	40.18	40.18	**40.19**
	4.0	37.10	40.65	38.33	40.53	38.96	35.63	40.52	40.63	40.69	40.70	40.70	**40.72**
lena	0.5	18.80	22.51	20.68	21.76	20.86	22.03	22.03	22.47	22.28	22.43	22.45	**22.53**
	1.0	23.18	26.41	23.22	26.33	23.97	26.17	26.17	26.39	26.48	26.51	26.57	**26.63**
	2.0	22.64	30.55	27.23	30.30	28.77	30.24	30.24	30.08	30.57	30.58	30.60	**30.68**
	3.0	26.44	32.18	27.89	32.13	29.35	31.79	31.79	32.10	32.27	32.26	**32.29**	32.26
	4.0	25.94	33.38	29.82	33.23	31.03	33.09	33.09	33.30	33.39	33.42	**33.44**	33.42

(b)

Samp. Density (%)	Mean Rank[a]											
	SE	Del.	ABN	JND	DLP	DP	AMC	GHH	SQSE	JNDSE	MASE	ELSE
0.5	10.98 (1.33)	4.40 (1.69)	9.68 (1.13)	7.65 (1.75)	8.95 (1.75)	11.00 (1.76)	6.15 (2.34)	5.60 (2.07)	4.60 (2.00)	4.33 (1.89)	2.98 (1.37)	**1.70** (1.36)
1.0	11.48 (0.59)	4.55 (1.18)	9.88 (0.60)	7.78 (0.79)	9.00 (0.89)	11.43 (0.54)	6.60 (1.61)	5.78 (1.25)	3.90 (1.04)	3.60 (1.43)	2.33 (0.96)	**1.70** (1.65)
2.0	11.33 (0.47)	5.05 (0.77)	9.83 (0.38)	7.25 (0.83)	9.18 (0.38)	11.68 (0.47)	7.28 (1.07)	6.05 (0.77)	3.58 (1.20)	3.18 (0.77)	2.48 (0.77)	**1.15** (0.65)
3.0	11.38 (0.48)	5.05 (0.38)	9.93 (0.26)	7.05 (0.67)	9.08 (0.26)	11.63 (0.48)	7.68 (0.85)	5.98 (0.57)	3.70 (0.81)	3.15 (0.91)	2.10 (0.54)	**1.30** (1.03)
4.0	11.38 (0.48)	5.00 (0.55)	9.98 (0.16)	7.03 (0.47)	9.03 (0.16)	11.63 (0.48)	7.78 (0.82)	5.95 (0.63)	3.68 (0.88)	3.05 (0.74)	2.18 (0.83)	**1.35** (0.91)
All	11.31 (0.77)	4.81 (1.06)	9.86 (0.62)	7.35 (1.05)	9.05 (0.91)	11.47 (0.94)	7.10 (1.59)	5.87 (1.21)	3.89 (1.31)	3.46 (1.32)	2.41 (0.99)	**1.44** (1.19)

[a] The standard deviation is given in parentheses.

very well) is also shown. First, let us consider the example in Figure 3.3. Examining Figure 3.3(b), we can see that the image-domain triangulation obtained with the SE criterion has a large number of poorly-chosen sliver triangles, which lead to very high error in the corresponding image approximation in Figure 3.3(a). In contrast, viewing Figures 3.3(c) and (d), we observe that the ELSE criterion does not suffer from this problem. Now, moving our attention to the second example in Figure 3.4, we can see that a similar pattern of behavior is obtained as in the first example. Again, the SE criterion yields a triangulation with many poorly-chosen sliver triangles, which severely degrades approximation quality.

As for why the SE criterion typically yields triangulations with many poorly-chosen sliver triangles, this can be attributed to the combination of two factors. First, the SE criterion does not explicitly consider triangle shape and, therefore, does not have any direct mechanism for preventing the creation of bad sliver triangles or eliminating such triangles once they are present. Second, the SE criterion is also unable to account for triangle shape in an indirect manner, due to the shortsightedness of the LOP and LLOP. (The shortsightedness of the LOP and LLOP follows from the fact that a decision made at any given step in each of these algorithms considers the impact of that decision only in the current step, not in *all subsequent* steps.) In practice, the above two factors conspire to produce a pattern of behavior with the SE criterion that resembles the following. When a new point is inserted in the triangulation, a sliver triangle will sometimes result. In such a case, since the SE criterion does not directly consider triangle shape, the SE criterion will often be unable to eliminate the sliver triangle. Thus, as more points are inserted in the triangulation, the number of sliver triangles tends to grow significantly. In turn, as the number of sliver triangles grows, the number of unflippable edges also tends to increase. This leads to sliver triangles tending to have fewer flippable edges (on average). This, in turn, makes it more difficult to eliminate sliver triangles, once present. In this manner, a very large number of sliver triangles is obtained. Because the number of sliver triangles produced is so abnormally large, it is not surprising that the number of such triangles that are poorly chosen is also high.

In the experimental results above, we saw that the ELSE and MASE criteria perform best in terms

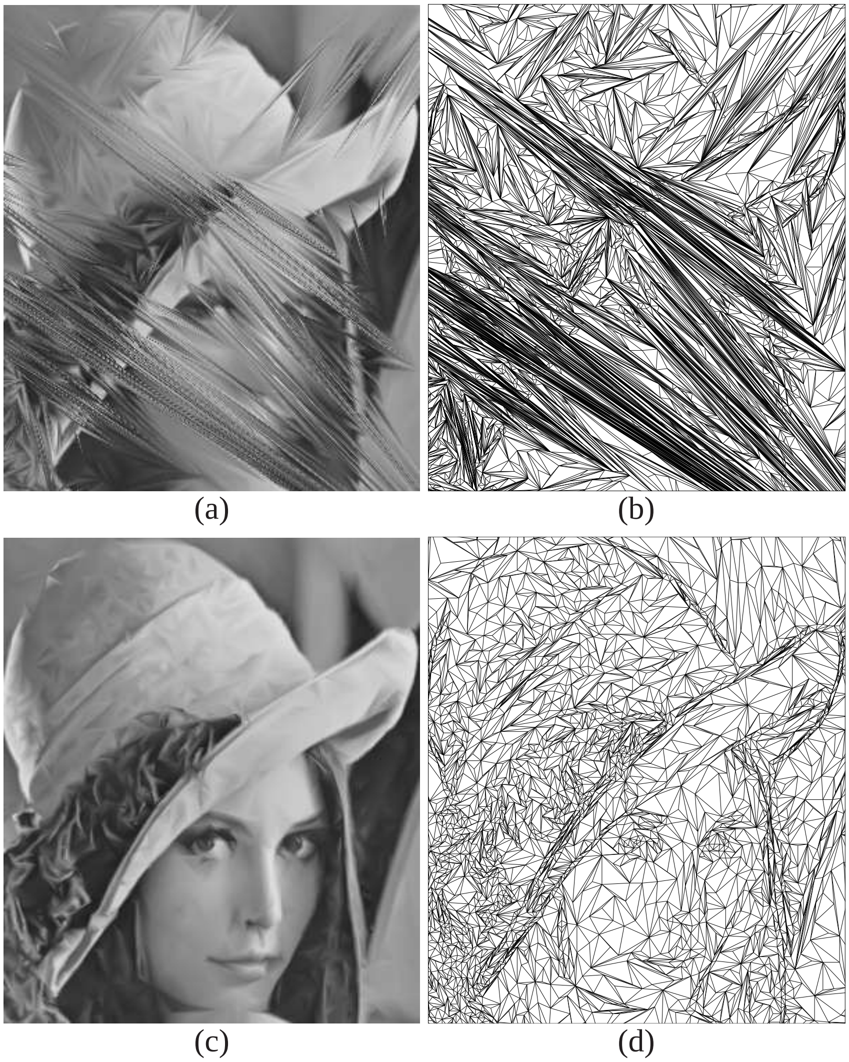


Figure 3.3: Comparison of the mesh quality obtained for the lena image at a sampling density of 2% in the case that the final-connectivity-adjustment method fcaMethod is LOP. Part of the image approximation obtained when the optimality criteria insOptCriterion is chosen as each of (a) SE (20.72 dB) and (c) ELSE (30.14 dB), and (b) and (d) the corresponding triangulations.

of mesh quality. This excellent performance is made possible by the fact that each of these two criteria has a direct dependence on *both* triangle shape and squared error. The dependence on squared error is critical for achieving high mesh quality, while the dependence on triangle shape is important for avoiding large numbers of poorly-chosen sliver triangles. In the case of the ELSE criterion, triangle

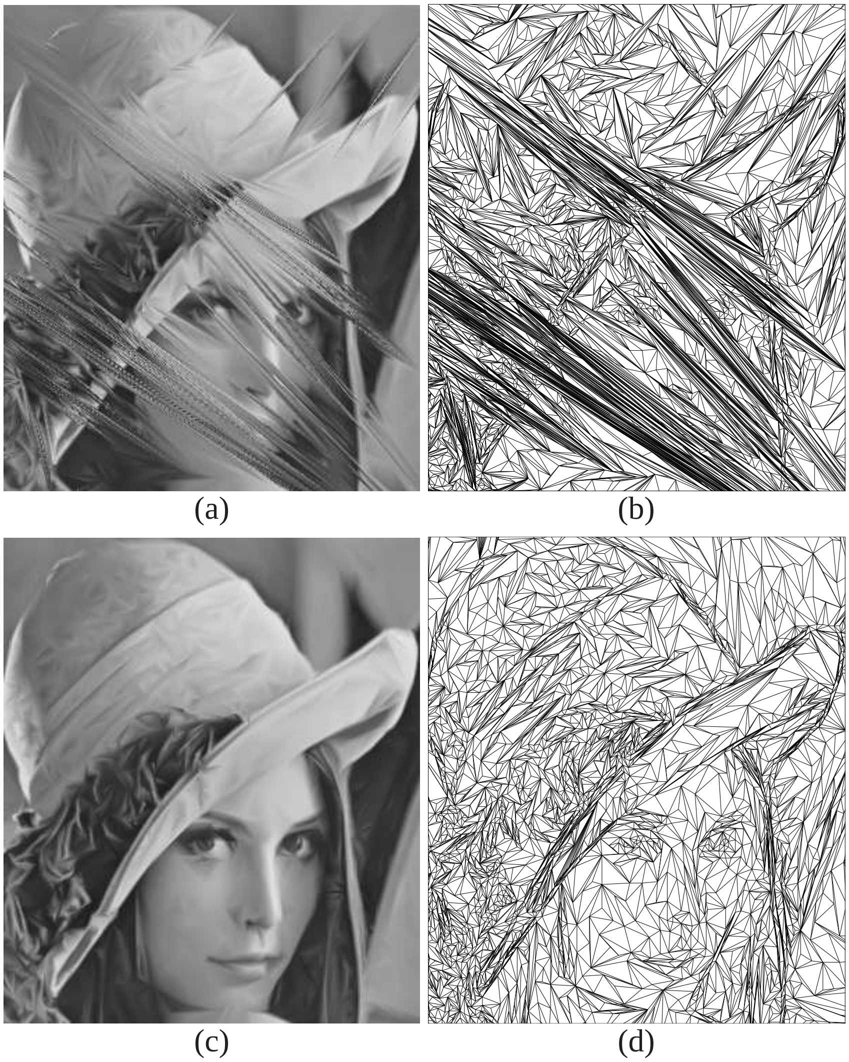


Figure 3.4: Comparison of the mesh quality obtained for the lena image at a sampling density of 2% in the case that the final-connectivity-adjustment method fcaMethod is LLOP. Part of the image approximation obtained when the optimality criteria insOptCriterion is chosen as each of (a) SE (22.64 dB) and (c) ELSE (30.68 dB), and (b) and (d) the corresponding triangulations.

shape is implicitly considered by the criterion's dependence on edge length, which penalizes longer edges. In the case of the MASE criterion, triangle shape is considered by the criterion's dependence on minimum interior angle, which penalizes smaller interior angles. By accounting for triangle shape, the ELSE and MASE criteria are able to avoid the bad-sliver problem that plagues the SE criterion.

Method for Final Connectivity Adjustment. Next, we study how the choice of the final-connectivity-adjustment method fcaMethod in step 3 of our framework (which can be either LOP or LLOP) affects mesh quality. To do this, we fix the insOptCriterion parameter to be ELSE and proceed as follows. For each of the 40 images in our test set and five sampling densities per image (for a total of $40 \cdot 5 = 200$ test cases), we generated a mesh using each of the two choices for fcaMethod under consideration (namely, LOP and LLOP), and measured the resulting approximation error in terms of PSNR. In all of these 200 test cases, the LLOP outperformed the LOP by a margin of 0.09 to 2.30 dB, with the average margin being 0.56 dB. Individual results for three images (namely, the images listed in Table 3.1) are given in Table 3.4. Examining this table, we see that the LLOP outperforms the LOP in all cases by a margin of 0.30 to 0.93 dB. Although we have only shown results for one choice of the fixed parameter insOptCriterion (i.e., ELSE), we found similar results with other choices. Thus, from above, we conclude that choosing the parameter fcaMethod as LLOP (as opposed to LOP) yields higher mesh quality. This said, however, we must point out that this choice entails a trade off in terms of computational cost. As noted earlier (in Section 2.9 on page 23), the LLOP has a higher computational cost than the LOP. For example, for the test case of the lena image at a sampling density of 2%, we found the LOP and LLOP to have computation times of about 1.40 seconds and 2.30 seconds, respectively. More generally, we have found the LLOP to typically require a computation time that is about 1.4 to 1.7 times that of the LOP. Thus, the best choice for the parameter fcaMethod depends on the most appropriate trade off between mesh quality and computational cost for the application at hand.

3.5 Proposed Methods

Above, we have considered how various choices for the free parameters in our computational framework for mesh generation (namely, the insertion order, triangulation optimality criterion, and final-connectivity-adjustment method) affect mesh quality. This led us to conclude that the triangulation optimality criterion insOptCriterion and the insertion order insOrder are best chosen as ELSE and

Table 3.4: Comparison of the mesh quality obtained with each of the two choices for the fcaMethod parameter

Image	Samp. Density (%)	PSNR (dB)	
		LOP	LLOP
bull	0.5	35.52	**36.20**
	1.0	29.99	**42.78**
	2.0	42.72	**43.51**
	3.0	43.97	**44.58**
	4.0	44.72	**45.33**
cr	0.5	35.11	**36.04**
	1.0	37.31	**37.82**
	2.0	39.10	**39.43**
	3.0	39.87	**40.19**
	4.0	40.42	**40.72**
lena	0.5	21.96	**22.53**
	1.0	26.13	**26.63**
	2.0	30.14	**30.68**
	3.0	31.72	**32.26**
	4.0	32.88	**33.42**

randomized order, respectively. Whether the final-connectivity-adjustment method fcaMethod should be chosen as LOP or LLOP is less clear cut, due to a trade off between mesh quality and computational cost. As a result, we chose to propose two methods, known as MED1 and MED2, where the first method has a lower computational cost relative to the second. The MED1 and MED2 methods both employ the best choices for insOptCriterion and insOrder as identified above (i.e., ELSE and randomized order, respectively). For the final-connectivity-adjustment method fcaMethod, however, the MED1 method uses the LOP (which has lower computational cost), while the MED2 method uses the LLOP (which has higher computational cost).

3.6 Evaluation of Proposed Methods

Having introduced our MED1 and MED2 mesh-generation methods, we now compare their performance to that of the ED scheme in terms of mesh quality. In addition, we make a few comments regarding the computational cost of our proposed methods. The software implementations of the methods used in this evaluation were developed by the author of this thesis and written in C++. For test data, we employ the same set of 40 images described earlier in Section 3.3.

Mesh Quality. For all 40 images in our test set and five sampling densities per image (for a total of $40 \cdot 5 = 200$ test cases), we used each of the various methods under consideration to generate a mesh, and then measured the resulting approximation error in terms of PSNR. Individual results for three specific images (namely, the images listed in Table 3.1) are given in Table 3.5.

To begin, we compare the MED1 and MED2 methods to the ED scheme. Examining the results for the individual test cases in Table 3.5, we see that the MED1 and MED2 methods both outperform the ED scheme in all 15 test cases, by margins of 1.94 to 8.46 dB and 2.24 to 9.14 dB, respectively. Next, we consider the full set of results taken across all 200 test cases (i.e., 40 images with five sampling densities per image). In the full set of results, we found that the MED1 and MED2 methods both yield higher quality meshes than the ED scheme in all 200 test cases. More specifically, the MED1 method outperformed the ED scheme by a margin of 1.93 to 8.46 dB with an average margin of 3.26 dB, while the MED2 method outperformed the ED scheme by a margin of 2.23 to 9.14 dB with an average margin of 3.81 dB. Thus, the MED1 and MED2 methods clearly yield meshes of very substantially higher quality, relative to the ED method.

Next, we compare the performance of the MED1 and MED2 methods. Examining the results from the individual test cases in Table 3.5, we can see that the MED2 method beats the MED1 method in all 15 test cases by a margin of 0.30 to 0.93 dB. In the full set of results, we found that the MED2 method yields higher quality meshes than the MED1 method in all 200 test cases, by a margin of 0.09 to 2.30 dB with an average margin of 0.56 dB. Therefore, the MED2 method consistently yields

Table 3.5: Comparison of the mesh quality obtained with the various methods

Image	Samp. Density (%)	PSNR (dB)		
		MED1	MED2	ED
bull	0.5	35.52	36.20	27.06
	1.0	39.99	40.78	34.46
	2.0	42.72	43.51	38.59
	3.0	43.97	44.58	40.47
	4.0	44.72	45.33	41.60
cr	0.5	35.11	36.04	31.96
	1.0	37.31	37.82	33.84
	2.0	39.10	39.43	35.72
	3.0	39.87	40.19	37.63
	4.0	40.42	40.72	38.48
lena	0.5	21.96	22.53	17.76
	1.0	26.13	26.63	21.50
	2.0	30.14	30.68	26.38
	3.0	31.72	32.26	28.50
	4.0	32.88	33.42	29.83

meshes of higher quality than the MED1 method. This behavior is due to the MED2 method using the more effective LLOP (instead of the LOP) for final connectivity adjustment.

In the above results, PSNR was found to correlate reasonably well with subjective image quality. For the benefit of the reader, however, we include illustrative examples in what follows. To begin, we consider the case of a photographic image. For one of the test cases in Table 3.5 namely, the lena image at a sampling density of 2%, part of the image approximation and the corresponding image-domain triangulation obtained for each of the various methods is shown in Figure 3.5. Examining this figure, we can see that the image approximations produced by our MED1 and MED2 methods (in Figures 3.5(a) and (b), respectively) are clearly of much higher quality than the one produced by the ED scheme (in Figure 3.5(c)), with image details such as image edges/contours being much better preserved in the MED1 and MED2 cases. In order to more clearly highlight some of the more subtle difference between the results for our MED1 and MED2 methods, we show (for the same test case) the results for a smaller region of interest under greater magnification in Figure 3.6. By carefully comparing the image approximations for our MED1 and MED2 methods in Figures 3.6(a) and (b), respectively, we can see that there are a few places where image details (such as edges and contours) are slightly better preserved by our MED2 method than our MED1 scheme, one example being the (image) edge along the top of the hat. The improved performance in the MED2 case is largely due to triangulation edges being better aligned with image edges/contours. So, in terms of subjective quality, our MED1 and MED2 methods are both vastly superior to the ED scheme, with our MED2 method yielding slightly better quality than our MED1 scheme.

Now, we consider the cases of computer-generated and medical images. In particular, for two test cases in Table 3.5, namely, the bull image at a sampling density of 1% and the cr image at a sampling density of 0.5%, part of the image approximations and the corresponding image-domain triangulations obtained for each of the various methods are shown in Figures 3.7 and 3.8, respectively. Observing the two figures, we can see that the trends in these two figures are similar to those in Figure 3.5. In particular, the image approximations for our MED1 and MED2 methods (in Figures 3.7(a) and 3.8(a), and 3.7(b) and 3.8(b), respectively) are clearly superior to those for the ED method (in Figures 3.7(c) and 3.8(c)), with image details (such as edges and contours) being much better preserved in the MED1 and MED2 cases.

Computational Cost. Next, we briefly consider the computational costs of our MED1 and MED2 methods. For the purposes of making timing measurements, we employed very modest hardware, namely an eight-year-old notebook computer with a 2.00 GHz Intel Core2 Duo T7250 CPU and 1.0 GB of RAM. On this machine, our MED1 and MED2 methods typically require only a few seconds of computation time for images like lena from Table 3.1. In particular, for the lena image and several sampling densities, the time required for mesh generation for each method under consideration is shown in Table 3.6.

Examining Table 3.6, we can see that the MED1 and MED2 methods require 0.95 to 2.06 seconds and 1.38 to 3.30 seconds, respectively. We can also observe that our MED1 and MED2 methods typically increase the computation time relative to the ED scheme by only 0.3 to 1.1 seconds and 0.7 to 2.4 seconds, respectively. In absolute terms, this incremental cost is very small when one

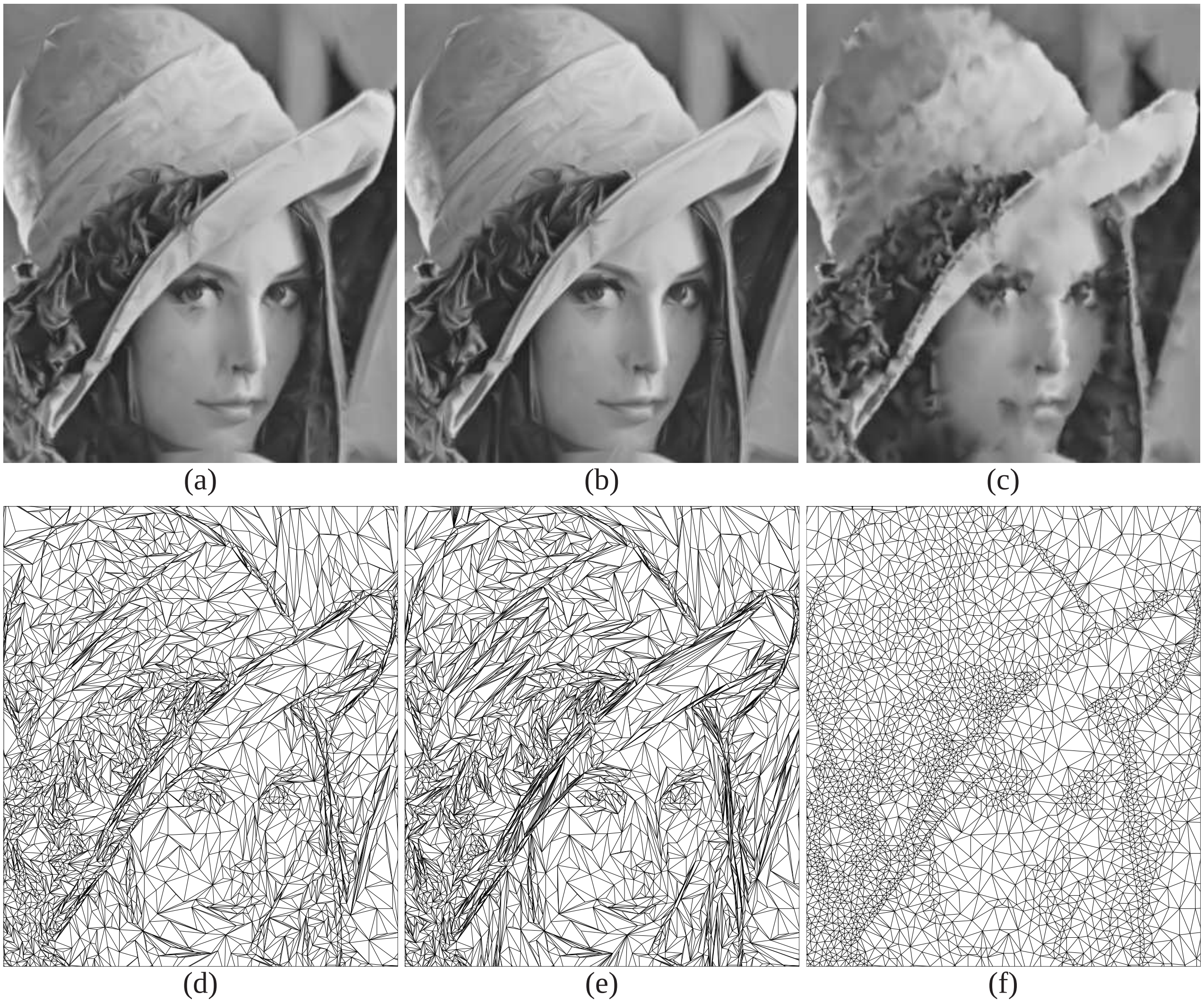


Figure 3.5: Part of the image approximation obtained for the lena image at a sampling density of 2% with each of the (a) MED1 (30.14 dB), (b) MED2 (30.68 dB), and (c) ED (26.38 dB) methods, and (d), (e), and (f) the corresponding triangulations.

considers the very substantial improvement in mesh quality obtained with our methods. Furthermore, when viewed in the broader context of the many mesh-generation techniques proposed to date in the literature, our MED1 and MED2 methods are quite low in terms of computational cost. For example, some other methods, which are based on techniques such as simulated annealing or simplification of very large meshes, can easily require computation times on the order of minutes or more.

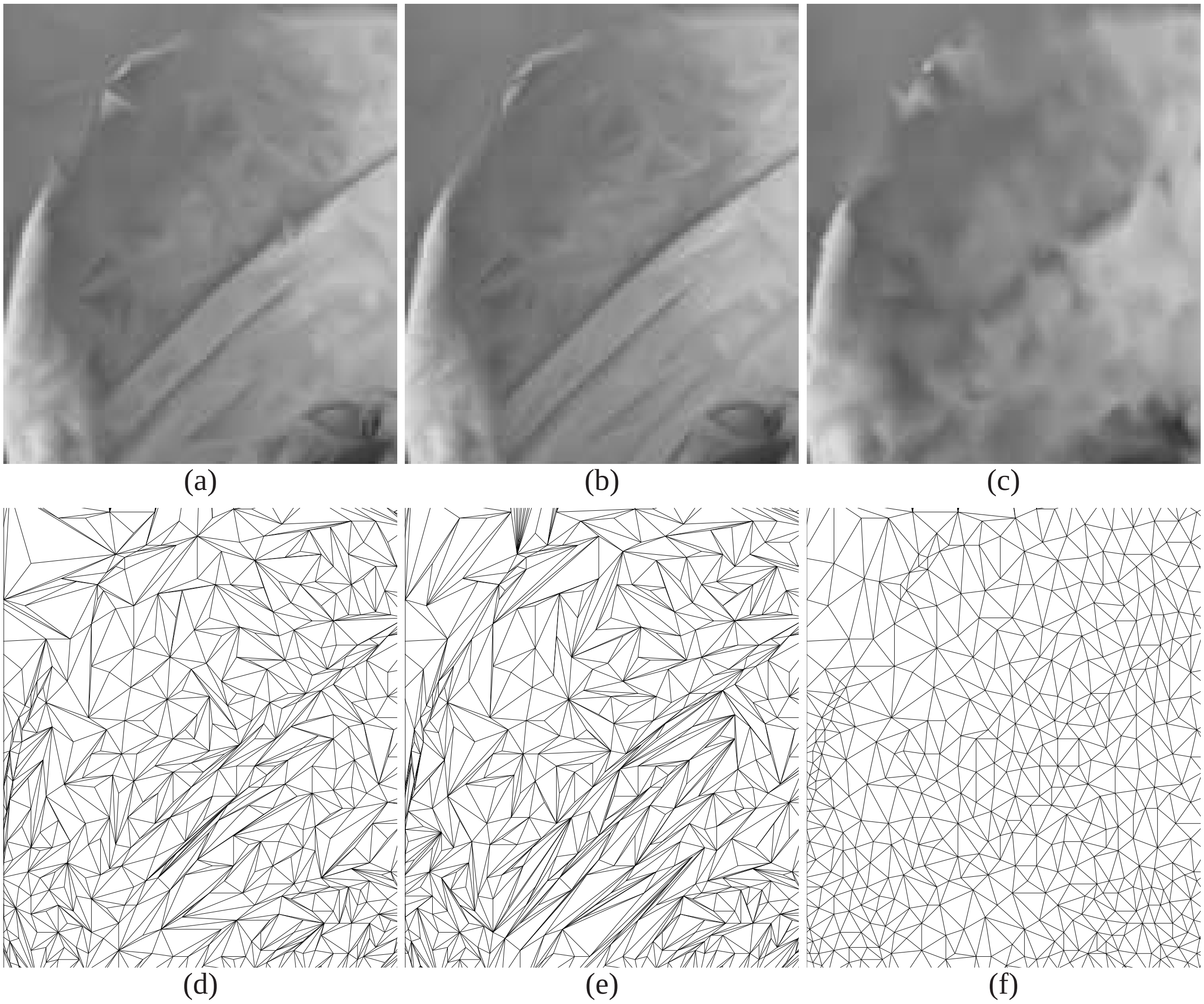


Figure 3.6: Part of the image approximation (under magnification) obtained for the lena image at a sampling density of 2% with each of the (a) MED1 (30.14 dB), (b) MED2 (30.68 dB), and (c) ED (26.38 dB) methods, and (d), (e), and (f) the corresponding triangulations.

Typically, the computation time for our MED2 method was found to be about 1.4 to 1.7 times that of our MED1 scheme (which is supported by the timing results in Table 3.6). So, our MED2 method is more computationally expensive, with this higher cost coming from the use of the LLOP (instead of the LOP) during final connectivity adjustment. As we saw earlier, our MED2 method yields higher quality meshes than our MED1 scheme. So, whether our MED1 or MED2 method is more attractive

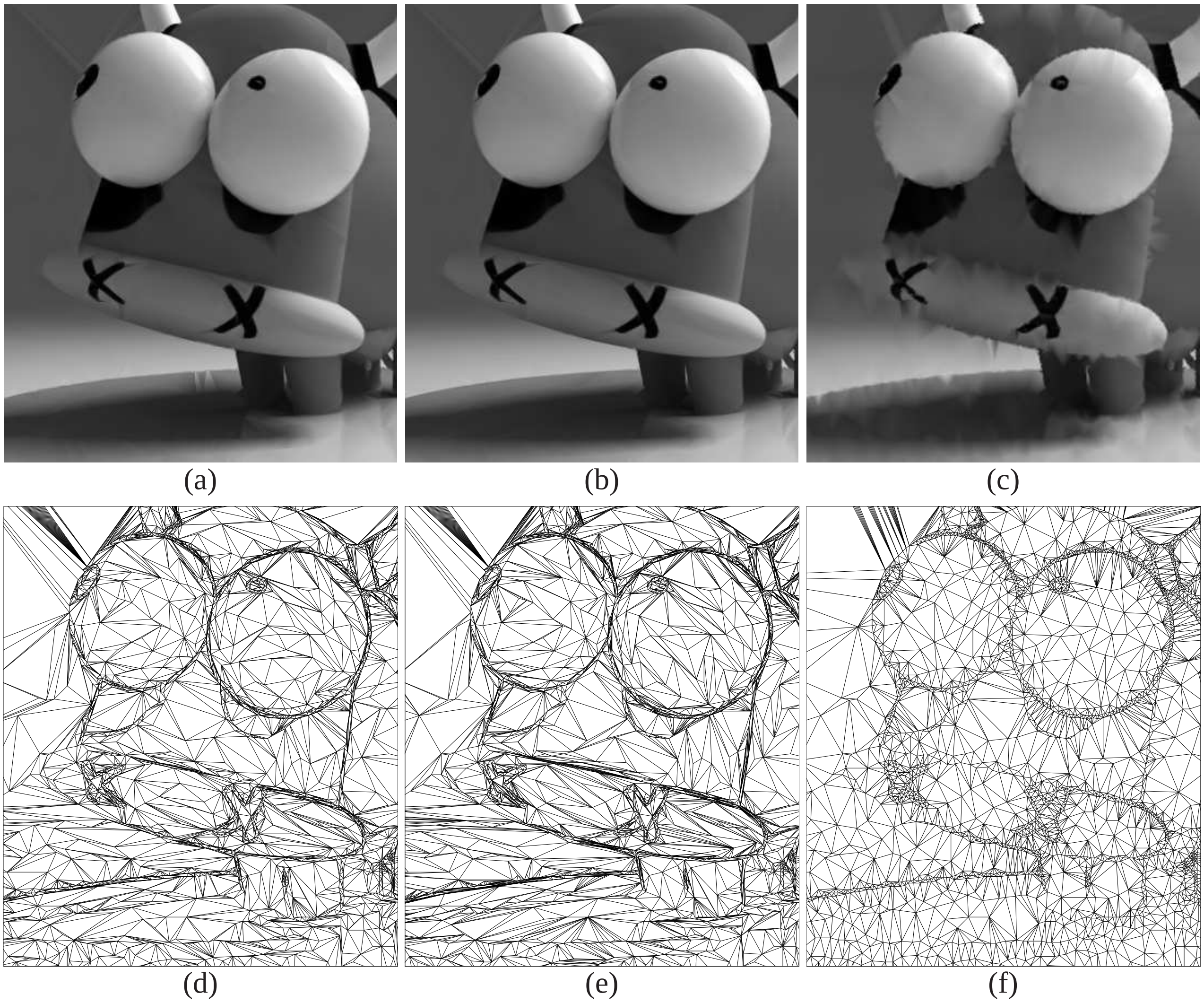


Figure 3.7: Part of the image approximation obtained for the bull image at a sampling density of 1% with the (a) MED1 (39.99 dB), (b) MED2 (40.78 dB), and (c) ED (34.46 dB) methods and (d), (e), and (f) the corresponding triangulations.

for a particular application, depends on computational constraints. Our MED1 method would be more appropriate for more computationally-constrained environments, whereas our MED2 method would be preferred otherwise.

Lastly, we must make one additional comment regarding the timing results in Table 3.6. As the astute reader might have noticed, the execution time required for the ED method in Table 3.6 does not

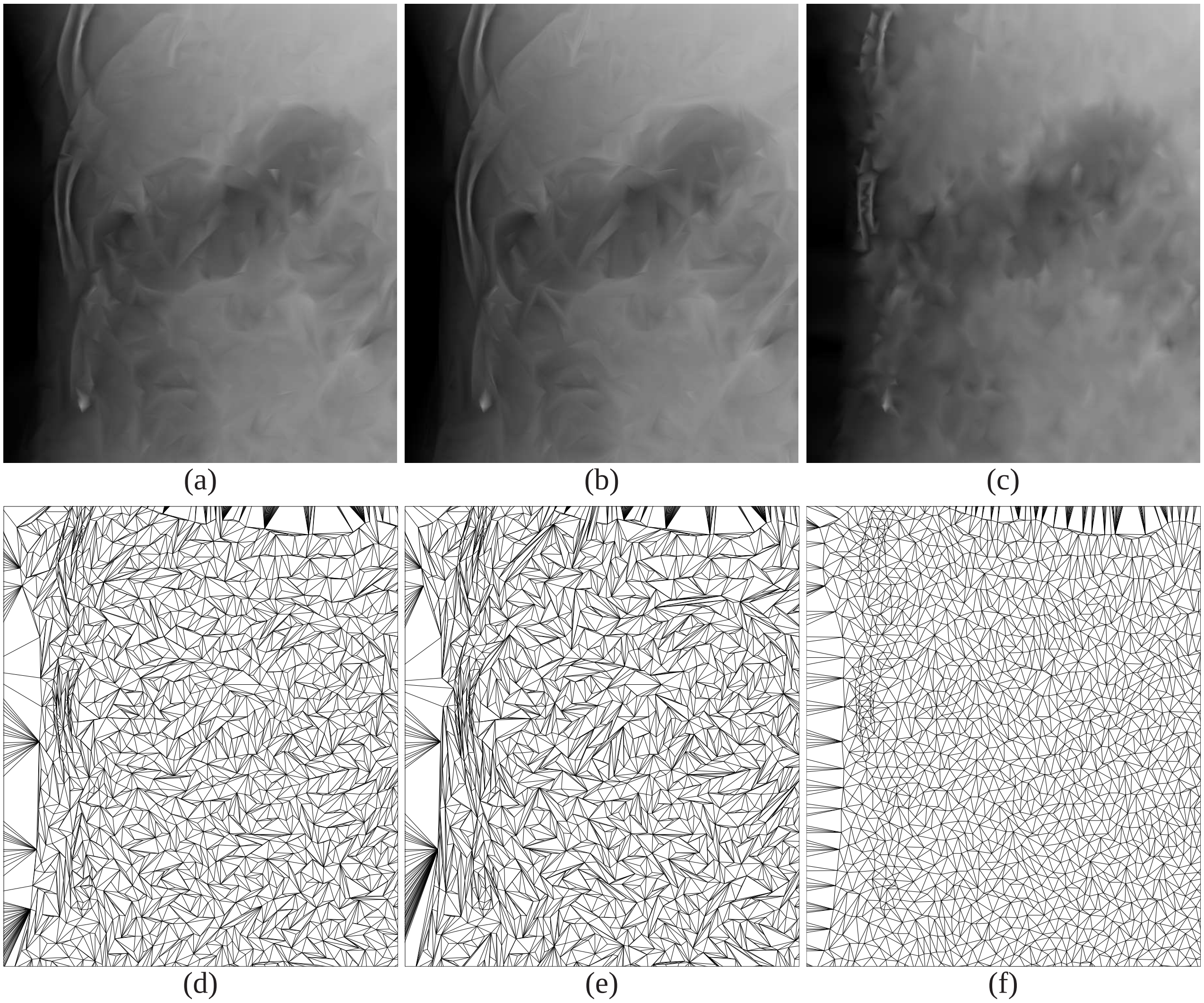


Figure 3.8: Part of the image approximation obtained for the cr image at a sampling density of 0.5% with the (a) MED1 (35.11 dB), (b) MED2 (36.04 dB), and (c) ED (31.96 dB) methods and (d), (e), and (f) the corresponding triangulations.

increase monotonically with the sampling density, namely, the time required for the sampling density of 1% is lower than that for 0.5%. We explain the reasons for this phenomenon as follows. Recall that in Section 2.7, we mentioned that the Floyd-Steinberg error diffusion is performed iteratively in our work in order to achieve exactly the desired number of sample points. Since the required number of iterations does not increase monotonically with the sampling density, it would not be unreasonable if

Table 3.6: Comparison of computational complexity for the various methods

Image	Samp. Density (%)	Time (s)		
		MED1	MED2	ED
lena	0.5	0.95	1.38	0.60
	1.0	1.03	1.68	0.51
	2.0	1.40	2.30	0.66
	3.0	1.76	2.81	0.82
	4.0	2.06	3.30	0.99

the time required for the sample-point selection decreases as the sampling density increases by a small percentage. Therefore, if the sample-point selection takes up a relatively large proportion of the total time required for a mesh-generation method (e.g, the ED method), it would not be unreasonable if the time required for mesh generation decreases as the sampling density increases by a small percentage. This is why for the ED method in Table 3.6, the time required for the sampling density of 1% is lower than that for 0.5%. This being said, however, the general trend is still that the time required for mesh generation increases with the sampling density (as supported by the timing results in Table 3.6).

3.7 Other Work

At this point, we have proposed two novel mesh-generation methods and evaluated their performance by comparing to the ED method in terms of mesh quality and computational complexity. During the development of our framework, we explored additional variations on our approach beyond what we have shown earlier. These other variations were found not to lead to better results, however. For this reason, we decided not to include discussions of those variations in the material presented earlier. For the sake of completeness, we briefly discuss those variations and show some extra experimental results in what follows.

3.7.1 Selection of Free Parameters for ED Method

Recall that in the ED method (as introduced in Section 2.7 on page 17), the sample points are selected in a way that they are distributed approximately proportional to the MMSODD of the image. In other words, we chose to use the MMSODD as the sample-point density function for Floyd-Steinberg error diffusion in the ED method. Also in Section 2.7, we only considered the variant of the ED method for which the serpentine scan order is used and the sensitivity parameter γ is chosen as 1. During the development of our work, we studied more choices of the above three parameters for the ED method, namely, the sample-point density function for Floyd-Steinberg error diffusion, the scan order for Floyd-Steinberg error diffusion, and the sensitivity parameter γ. Those additional choices are described as follows:

1. Both the MMSODD (as given by (2.1) on page 8) and skewness (as given by (2.2) on page 9) were explored as the choice of the sample-point density function for Floyd-Steinberg error diffusion.

2. Both the raster and serpentine scan orders from [19] were explored for Floyd-Steinberg error diffusion.

3. Three values were studied for the sensitivity parameter γ, namely 0.5, 1, and 2.

It is worth noting that, during our experimentation, the other characteristics of the ED method are fixed as follows:

1. a third-order binomial filter is used for smoothing;

2. non-leaky error diffusion is used; and

3. the error diffusion algorithm is performed iteratively in order to achieve exactly the desired number of sample points.

Table 3.7: Comparison of different sample-point density functions for the ED method

Image	Samp. Density (%)	PSNR (dB)	
		Skewness	MMSODD
bull	0.50	16.77	**27.06**
	1.00	16.94	**34.46**
	2.00	18.34	**38.59**
	3.00	18.79	**40.47**
	4.00	19.28	**41.60**
cr	0.50	15.35	**31.96**
	1.00	20.88	**33.84**
	2.00	21.42	**35.72**
	3.00	22.97	**37.63**
	4.00	23.24	**38.48**
lena	0.50	15.13	**17.76**
	1.00	16.26	**21.50**
	2.00	17.73	**26.38**
	3.00	19.07	**28.50**
	4.00	19.55	**29.83**

In what follows, we proceed to show how we arrived at a particular choice for each of the three parameters (i.e., the sample-point density function, the scan order, and the sensitivity parameter γ) for the ED method with some experimental results.

Sample-Point Density Function. To begin, we studied how different choices of the sample-point density function used for Floyd-Steinberg error diffusion in the ED method affect mesh quality. We fixed the scan order to be serpentine scan order and the sensitivity parameter γ to be 1, and proceeded as follows. For numerous images and sampling densities, we generated a mesh employing the ED method with the sample-point density function chosen as each of the MMSODD and skewness, and measured the resulting mesh quality in terms of PSNR. A representative subset of the results is shown in Table 3.7. From this table, we can observe that the MMSODD yields meshes of substantially higher quality than those yielded with the skewness. In particular, the MMSODD beats the skewness in all 15 test cases, by a margin of 2.63 to 22.32 dB.

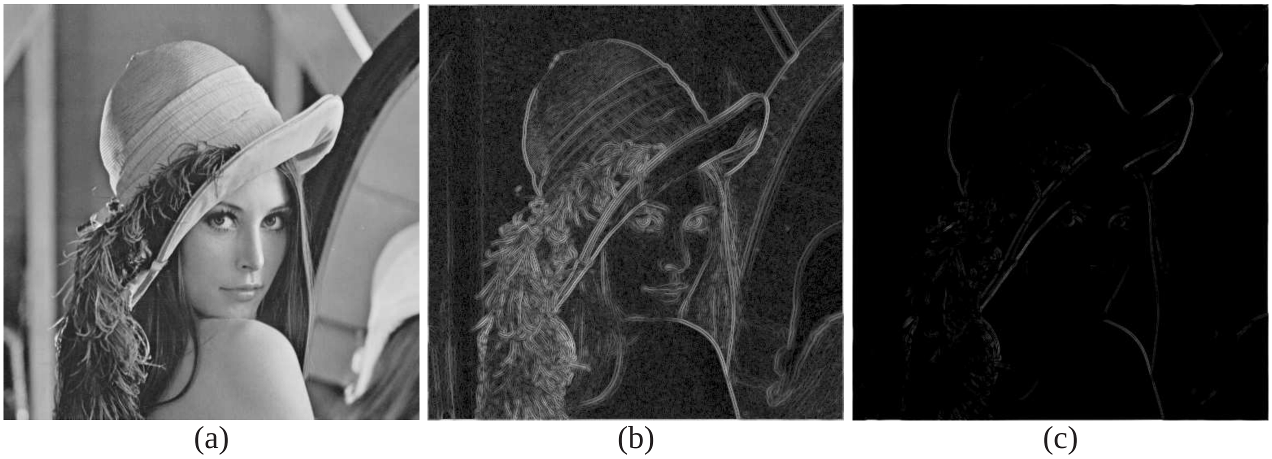

(a) (b) (c)

Figure 3.9: (a) An image ϕ, (b) the MMSODD of ϕ , and (c) the skewness of ϕ.

To help the reader understand why there is such a substantial difference between the meshes yielded with the use of the MMSODD and skewness, we provide an illustrative example. For the lena image (listed in Table 3.1), the MMSODD and skewness of the image are shown in Figures 3.9(b) and (c), respectively. Observing the MMSODD of the lena image as shown in Figure 3.9(b), we can see that many high-MMSODD pixels in the image tend to cluster around image edges and contours. For the skewness of the lena image as shown in Figure 3.9(c), however, we observe much less clustering of high-skewness pixels around image edges and contours. Since the skewness is clearly inferior to the MMSODD in terms of detecting image features (edges/contours) as shown above, we did not consider the use of the skewness in the material presented before Section 3.7.

Scan Order. Next, we studied how different choices of the scan order used for Floyd-Steinberg error diffusion in the ED method affect mesh quality. To clarify, the scan order refers to the order in which the grid points are visited during Floyd-Steinberg error diffusion. We fixed the sample-point density function to be the MMSODD and the sensitivity parameter γ to be 1, and proceeded as follows. For numerous images and sampling densities, we generated a mesh employing the ED method with the scan order chosen as each of the raster and serpentine scan orders, and measured the resulting mesh quality in terms of PSNR. A representative subset of the results is shown in Table 3.8. Examining Table 3.8, we can see that the serpentine scan order consistently outperforms the raster

Table 3.8: Comparison of different scan orders for the ED method

Image	Samp. Density (%)	PSNR (dB)	
		Raster	Serpentine
bull	0.50	25.63	**27.06**
	1.00	32.93	**34.46**
	2.00	38.04	**38.59**
	3.00	40.14	**40.47**
	4.00	41.46	**41.60**
cr	0.50	30.83	**31.96**
	1.00	32.74	**33.84**
	2.00	35.19	**35.72**
	3.00	37.00	**37.63**
	4.00	38.04	**38.48**
lena	0.50	17.07	**17.76**
	1.00	20.00	**21.50**
	2.00	24.62	**26.38**
	3.00	27.32	**28.50**
	4.00	28.98	**29.83**

scan order. In particular, the serpentine scan order beats the raster scan order in all 15 test cases, by a margin of 0.14 to 1.76 dB. Consequently, we did not consider the raster scan order in the material presented before Section 3.7.

Sensitivity Parameter γ. Lastly, we studied how different choices of the sensitivity parameter γ used for the ED method affect mesh quality. We fixed the sample-point density function to be the MMSODD and the scan order to be the serpentine scan order, and proceeded as follows. For numerous images and sampling densities, we generated a mesh employing the ED method with the scan order sensitivity parameter γ chosen as each of 0.5, 1, and 2, and measured the resulting mesh quality in terms of PSNR. A representative subset of results is shown in Table 3.9. From this table, we can see that the mesh produced in the case of $\gamma = 1$ consistently outperforms the meshes generated in the other two cases. In particular, the mesh produced in the case of $\gamma = 1$ beats the mesh generated in the case of $\gamma = 0.5$ in 13/15 of the test cases by a margin of 0.90 to 7.33 dB, and beats the mesh generated in the case of $\gamma = 2$ in all 15 test cases by a margin of 5.62 to 20.34 dB. Therefore, we only

Table 3.9: Comparison of different γ parameter choices for the ED method

Image	Samp. Density (%)	PSNR (dB)		
		$\gamma = 0.5$	$\gamma = 1$	$\gamma = 2$
bull	0.50	24.73	**27.06**	16.53
	1.00	33.30	**34.46**	17.26
	2.00	37.53	**38.59**	19.80
	3.00	39.07	**40.47**	20.73
	4.00	40.01	**41.60**	21.26
cr	0.50	24.63	**31.96**	24.07
	1.00	29.55	**33.84**	27.56
	2.00	34.50	**35.72**	30.10
	3.00	35.73	**37.63**	31.58
	4.00	36.26	**38.48**	32.82
lena	0.50	**19.23**	17.76	15.21
	1.00	**22.38**	21.50	16.41
	2.00	25.39	**26.38**	17.94
	3.00	27.33	**28.50**	18.91
	4.00	28.93	**29.83**	19.74

considered $\gamma = 1$ in the material presented before Section 3.7.

3.7.2 Other Extra Experiments

Above, we discussed the extra exploration of several free parameters for the ED method. In addition to the extra results shown in previous section, we studied some other parameters and even a few variants of the framework during the development of our framework. In what follows, we briefly discuss them for the reader's interest.

Attempt to Improve Performance of SE Criterion. In Section 3.4, we showed experimental results (in Figures 3.3 and 3.4) to illustrate that choosing the optimality criterion insOptCriterion in our framework as SE would lead to extremely poor image approximations, with many poorly-chosen sliver triangles being produced in the triangulation. During the development of our work, in order to improve the performance of the SE criterion, we tried a variant of our framework that employs the LLOP (instead of the LOP) for selecting triangulation connectivity after each point is inserted in the triangulation. Since the LLOP is often able to find a better local optimum than the LOP (as mentioned in Section 2.9 on page 23), we expected to obtain better results for the SE criterion with the use of the LLOP during point insertion. To evaluate the performance of the above variant of the our framework, we fixed the optimality criterion insOptCriterion to be SE and skipped the final-connectivity-adjustment step, and proceeded as follows. For numerous images and sampling densities, we generated a mesh with each of the LOP and LLOP being employed for triangulation connectivity adjustment during point insertion, and measured the resulting mesh quality in terms of PSNR. A representative subset of results is shown in Table 3.10. For comparison purposes, we have also included the results of our proposed MED1 and MED2 methods for the same test cases in Table 3.10. From this table, we can see that the LLOP yields meshes of much better higher quality than the LOP. In particular, the LLOP beats the LOP in all 15 test cases by a margin of 2.43 to 12.26 dB. Therefore, if the LLOP is used for triangulation connectivity adjustment during point insertion, the SE criterion is able to perform much better compared to the case that the LOP is employed. This being said, however, we must note that the quality of meshes yielded with the above variant (i.e., employing the LLOP during point insertion) is still not competitive with our proposed MED1 and MED2 methods, as shown in Table 3.10. For this reason, we did not consider the use of the LLOP during point insertion in the material presented before Section 3.7.

In the above evaluations, PSNR was found to correlate reasonably well with the subjective quality. We provide an example herein illustrating visual quality. For one of the test cases from Table 3.10 (namely, the lena image at a sampling density of 2%), Figure 3.10 shows part of the approximation and corresponding image-domain triangulation with each of the LOP and LLOP being employed for triangulation connectivity adjustment during point insertion. Examining Figure 3.10, we can see that the image approximation yielded in the case of the LLOP (in Figure 3.10(c)) is clearly of higher visual quality than the one yielded in the case of the LOP (in Figure 3.10(a)). Examining the image-domain triangulations for the cases of the LOP and LLOP (in Figures 3.10(b) and (d), respectively), we can easily determine the reason for the better approximation yielded in the case of the LLOP. That is,

Table 3.10: Comparison of the various mesh-generation methods

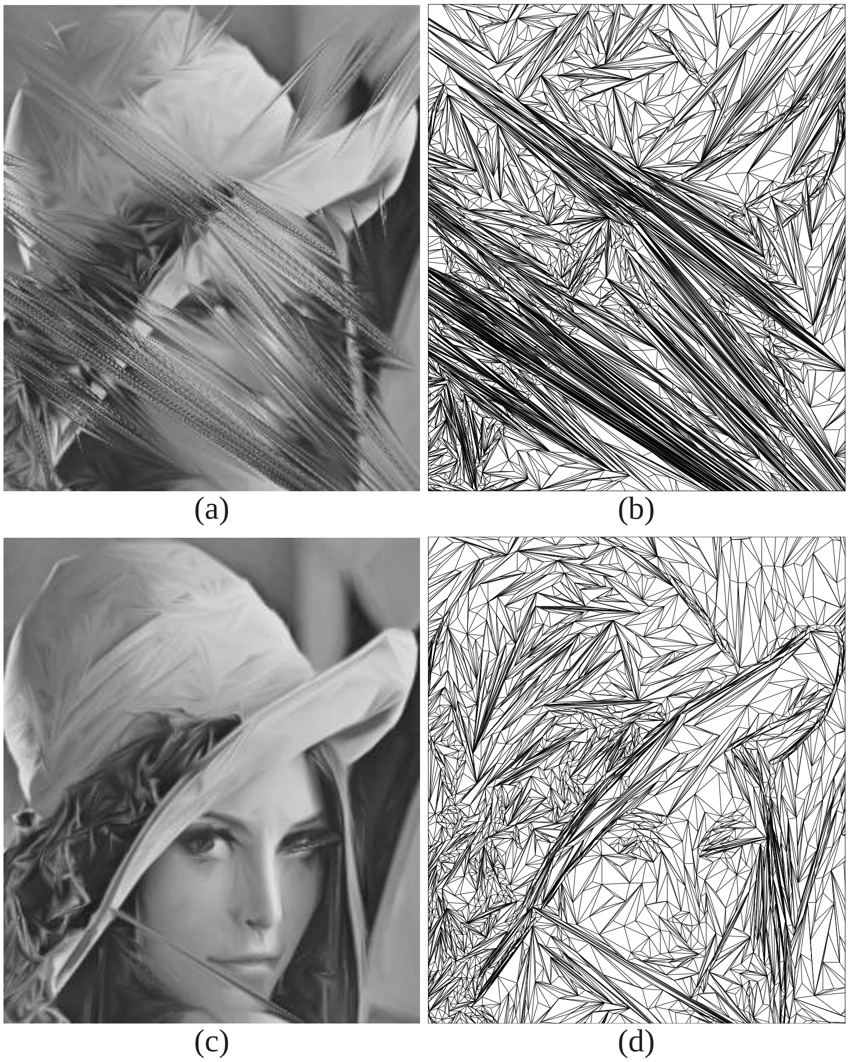
Image	Samp. Density (%)	PSNR (dB)			
		LOP[a]	LLOP[b]	MED1	MED2
bull	0.5	24.73	34.11	35.52	36.20
	1.0	26.89	36.42	39.99	40.78
	2.0	30.53	39.19	42.72	43.51
	3.0	31.93	43.09	43.97	44.58
	4.0	31.23	43.49	44.72	45.33
cr	0.5	31.19	33.62	35.11	36.04
	1.0	32.41	36.59	37.31	37.82
	2.0	33.33	38.28	39.10	39.43
	3.0	34.12	39.18	39.87	40.19
	4.0	35.63	39.48	40.42	40.72
lena	0.5	17.61	20.49	21.96	22.53
	1.0	21.50	24.69	26.13	26.63
	2.0	20.72	28.94	30.14	30.68
	3.0	23.43	30.26	31.72	32.26
	4.0	23.67	31.89	32.88	33.42

[a] The LOP is used during point insertion.

[b] The LLOP is used during point insertion.

using the LLOP for triangulation connectivity adjustment during point insertion has helped eliminate many poorly-chosen sliver triangles caused by the SE criterion (which the LOP was not able to do), leading to better image approximations.

Extra Optimality Criteria. Recall that in Section 3.2, we considered twelve possibilities for the choice of the optimality criterion insOptCriterion in our framework, two of which (i.e., ELSE and MASE) are newly proposed in this thesis. During the development of our work, we tried even more optimality criteria for the choice of insOptCriterion. Those new criteria we tried were mostly heuristic based, for which we explored the combination of the SE criterion (which considers approximation error) with another criterion that measures some geometric property of a mesh model such as the angle between normals (ABN). Unfortunately, none of the new heuristic-based criteria that we studied were competitive with the ELSE and MASE criteria, so they were not included in the material presented



(a) (b)

(c) (d)

Figure 3.10: Part of the triangulations obtained for the lena image at a sampling density of 2% with the SE criterion being used for the (a) LOP (20.72 dB) and (c) LLOP (28.94 dB) during point insertion, and (b) and (d) their corresponding triangulations.

before Section 3.7.

Two-Level Final Connectivity Adjustment. Recall that in step 3 of our framework (as introduced in Section 3.2 on page 27), we chose between the LOP and LLOP for the use of final connectivity adjustment. During the development of our work, we also explored a variant of the framework that employs two levels of final connectivity adjustment. That is, all the steps are the same with our proposed framework, except that after all the sample points are inserted in the triangulation, the LOP or LLOP is invoked two times, with different optimality criteria being employed, to adjust the final connectivity of the mesh. The results of the above variant were found to be highly data dependent. No specific optimality criterion could be advocated for each of the two LOPs or LLOPs used for final connectivity adjustment, as no particular choice of these criteria could lead to a method that was consistently better than others. For this reason, we did not consider the two-level final connectivity adjustment in the material presented before Section 3.7.

Chapter 4

Conclusions and Future Research

4.1 Conclusions

In this thesis, we have studied DDT-based triangle mesh models for image representation. In particular, we have proposed a computational framework for mesh generation that modifies the ED method to use DDTs in conjunction with the LOP, and derived two mesh-generation methods from this framework. By using DDTs in conjunction with the LOP (instead of Delaunay triangulations), triangulation connectivity can be chosen optimally so as to minimize approximation error. As the proposed framework has several free parameters, we studied how various choices for each of the free parameters affect the mesh quality and advocated two particular choices for those parameters, leading to two specific mesh-generation methods known as MED1 and MED2. Through experimental results, our MED1 and MED2 methods were shown to produce image approximations of much higher quality than the ED method, both in terms of PSNR and subjective quality, at a relatively modest computational cost. In particular, the MED1 and MED2 methods were shown to outperform the ED method by margins of approximately 3.26 to 3.81 dB on average. Our two methods allow a different trade off to be made between computational cost and approximation quality, allowing our approach to be useful in a wider range of applications. As part of our work, we proposed two novel optimality criteria to be used in conjunction with the LOP, namely the ELSE and MASE criteria. These two criteria were shown to outperform other well known criteria from the literature. Of our two newly proposed criteria, the ELSE criterion was found to perform best and was used as a key component of our MED1 and MED2 methods. Since the LOP is used in many different applications and the MSE (as in (2.3)) is a frequently employed error metric, our proposed ELSE and MASE criteria have the potential to be useful in a much broader range of contexts than the particular mesh-generation methods proposed herein. The MED1 and MED2 methods are of great utility to applications that employ mesh models of images. Furthermore, our new optimality criteria, ELSE and MASE, can be exploited by future mesh-generation schemes that employ the LOP in order to achieve improved results.

4.2 Future Research

Although the work presented in this thesis has covered a fairly wide range of topics related to DDT-based mesh-generation methods, there are still some potential areas that are worth exploring for future research. In what follows, we discuss these potential future research areas.

When presenting the LLOP in Section 2.9, we mentioned that one key way in which the LLOP differs from the LOP is that the LLOP allows two types of triangulation transformation. Instead of only allowing the triangulation to be transformed by a single edge flip in each step (as in the case of the LOP), the LLOP allows the triangulation to be transformed by: 1) a single edge flip; or 2) a sequence of two edge flips. Consequently, the LLOP is usually able to find a better local optimum than the LOP. This has been supported in the comparison of our two proposed methods, namely MED1 and MED2. It was shown in Section 3.6 that our MED2 method yields meshes of higher quality than our MED1 scheme, with the higher quality coming from the use of the more effective LLOP (instead of the LOP) for final connectivity adjustment. The above fact motivates us to think about the following possibility. If we introduce a third type of triangulation transformation in addition to the above two, by allowing a sequence of three edge flips, it is highly likely that we would be able to obtain even better results than the LLOP. One potential weakness of introducing a third type of transformation would be the extra computation being added relative to the LOP/LLOP. If an application has a less computationally-constrained environment and the boost in mesh quality is a high priority, however, this idea is still worth consideration.

One area that we have dedicated significant time exploring during the development of our work is the triangulation optimality criterion used for the LOP. Although we have proposed two novel optimality criteria (i.e., ELSE and MASE) that were shown to outperform other well known criteria from the literature, the future development of new high-performance optimality criteria can still be of great interest. We suggest two potential areas that can be studied during the development of new optimality criteria in the future. The first potential area is the development of new optimality criteria that are heuristic based. We have shown in Section 3.4 that heuristic-based criteria can lead to higher quality image approximations than cost-based criteria due to the fact that heuristic-based criteria take into account both triangle shape and squared error. Given the fact that our proposed ELSE and MASE criteria are both heuristic based and both perform extremely well, heuristic-based criteria are worth studying further. The second potential area is the development of better optimality criteria that employ the SE criterion. It was shown in Section 3.4 that the SE criterion performs extremely poorly when used during point insertion, mainly due to the existence of many poorly-chosen sliver triangles in the triangulation. Although we provided explanations on why the SE criterion typically yields triangulations with many poorly-chosen sliver triangles, more extensive investigation could be done in this regard for future work, with the goal of finding more specific reasons that caused the SE criterion to become trapped in such a poor local optimum. If more insightful reasons can be identified, they could potentially lead to better optimality criteria that employ the SE criterion and at the same time are capable of combating poorly-chosen sliver triangles effectively.

Appendix A

Software User Manual

A.1 Introduction

The software that implements the mesh-generation framework and methods proposed in the thesis was independently developed by the author, with a great deal of help from his supervisor, Dr. Michael Adams. The software was written in C++ to run under Linux, and consists of around 9000 lines of code. It makes heavy use of the Computational Geometry Algorithm Library (CGAL) [61] and the Signal Processing Library (SPL) [62], and also utilizes the Boost Library [63] and the SPL Extensions Library (SPLEL).

The software package contains two executable programs, `generate_mesh` and `reconstruct_image`, for mesh generation and image reconstruction, respectively. The `generate_mesh` program reads an image from a specified file and generates a mesh model of the image based on the desired sampling density and mesh-generation method specified by the user. The `reconstruct_image` program reads a mesh in a special format (to be introduced shortly) and produces a rasterized image.

The remainder of this appendix provides details on how to build and use the software. A few examples are also provided for illustrating how to use the software.

A.2 Building the Software

We chose to use the ***Make*** utility to build our software. Make is a utility that automatically builds executable programs and libraries from source code by reading files called makefiles which specify how to generate the target programs. Since some functionalities of C++11 are utilized in the code, the compiler one uses to compile the code has to be compatible with C++11. The compiler we have been using in our development is GCC 4.7.3. As mentioned earlier, our software package also utilizes libraries such as CGAL, Boost, SPL, and SPLEL. Thus, these libraries must be correctly installed before one can build our software. The versions of the libraries that have been verified to work with our software are:

- CGAL 3.8.2,
- Boost 1.46.0,
- SPL 1.1.11, and
- SPLEL 1.1.18.

To build the software, one needs to first go to the directory which contains the makefile. Then, to delete all the object files and executable files generated from previous building processes, run the command:

```
make clean
```

To generate the executable programs, invoke the command:

```
make
```

A.3 File Formats

The input of the `generate_mesh` program and the output of the `reconstruct_image` program are both **portable gray map (PGM)** [64] images. The mesh generated by the `generate_mesh` program is output in **object file format (OFF)** [65]. The input file of the `reconstruct_image` has a special format consisting of the following fields (in order), with each field separated by a white space or newline:

1. the width of the image,
2. the height of the image,
3. the maximum value of the image, and
4. the mesh of the image in OFF format.

We assign an extension ".mesh" to files with the above format.

For the benefit of the reader, we provide an example illustrating the above format. Suppose that we have an image with the width, height, and maximum value being 10, 10, and 255, respectively. The mesh model of the image is associated with a triangulation consisting of two triangles, namely ABD and BCD, as shown in Figure A.1, where vertices A, B, C, and D have xy-coordinates $(0,0)$, $(10,0)$, $(10,10)$, and $(0,10)$, respectively. The z values of A, B, C and D are 100, 200, 150, and 50, respectively. The ".mesh" file of the above mesh model would be as follows:

```
10 10 255
OFF
4 2 0
0 0 100
10 0 200
10 10 150
```

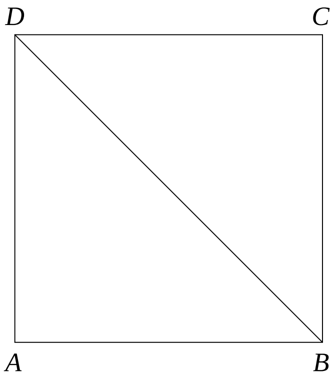


Figure A.1: A simple triangulation consisting of two faces.

```
0 10 50
3 0 1 3
3 1 2 3
```

A.4 Detailed Program Descriptions

As introduced earlier, our software contains two executable programs, namely `generate_mesh` and `reconstruct_image`. In what follows, we provide details in how to use these two programs.

A.4.1 The `generate_mesh` Program

SYNOPSIS

```
generate_mesh [OPTIONS]
```

DESCRIPTION

This program reads an image in PGM format from a specified file, and generates a mesh model of the image based on the desired sampling density and mesh-generation method specified by the user. The image file and desired sampling density are specified by the options `-f` and `-D`, respectively, and these two options are required by the program. With command-line options, the user can choose different sample-point-selection methods, different initial-mesh-construction methods, and different final-connectivity-adjustment methods. The width, height, and maximum value of the image, along with the generated mesh in OFF format are printed to standard output (by default). In the context of this appendix, we use the term ***maximum visit count*** to denote the maximum number of times that each edge is allowed to be tested for optimality in the LOP, and the term ***mesh size tolerance*** to represent the maximum difference allowed between the target size and actual size of the mesh model in the sample-point-selection process. The program returns 0 for a normal exit a nonzero value otherwise.

OPTIONS

`-f $inputImage` Specifies the image for processing to be `$inputImage`. This option is required.

`-D $sampDensity` Specifies the sampling density to be `$sampDensity` (in decimals). This option is required.

`-O $filtOrder` Specifies the order of binomial filter applied to the image to be `$filtOrder`. The default value of `$filtOrder` is 1 (no filtering).

`-g $gamma` Specifies the gamma parameter in the ED method to be `$gamma`. The default value of `$gamma` is 1.

`-S $subMethod` Chooses the method for sample point selection to be `$subsetMethod`, which can be chosen from Table A.2. The default value of `$subsetMethod` is ”ywb”.

`-z $szErrThresh` Specifies the mesh size tolerance to be `$szErrThresh`. The default value of `$szErrThresh` is 0.

`-I $insertCri` Chooses the optimality criterion for the LOP during point insertion to be `$insertCri`, which can be chosen from Table A.3. The default value of `$insertCri` is “delaunay”.

`-s $subsetFile` Specifies the file to which print the coordinates of the selected sample points. If not specified, the program will not output this information.

`-l $lookAhead` A flag to indicate whether the LLOP is enabled, “1” for enabled and “0” for disabled. By default the flag is “0”. Note that only the SE criterion is available for the LLOP.

`-c $maxVisits` Sets the maximum visit count to be `$maxVisits`. When an edge has been visited more than `$maxVisits` times, it is deemed that a cycle has occurred. The default value of `$maxVisits` is 10000.

`-v $exitOnCycle` A flag to indicate the special action to be taken when the maximum visit count has been exceeded for an edge. If `$exitOnCycle` is “0” (which is the default case), the program will skip this edge in the remainder of the currently running LOP; if `$exitOnCycle` is “1”, the program will terminate the currently running LOP immediately.

Table A.2: Various methods for sample point selection

Method	Description
ywb	MMSODD as density function and raster scan order in error diffusion
ywbs	MMSODD as density function and serpentine scan order in error diffusion
sked	skewness as density function and raster scan order in error diffusion
skeds	skewness as density function and serpentine scan order in error diffusion

`-m $mainProc` A flag to indicate whether the final-connectivity-adjustment step is enabled, with “1” for enabled and “0” for disabled. The default value of `$mainProc` is “1”.

`-E $mainCri` Chooses the optimality criterion for the final connectivity adjustment to be `$mainCri`, which can be chosen from Table A.3. The default value of `$mainCri` is “se”.

`-i $iviewerFile` Specifies a file to which print the generated triangulation in iviewer-readable format. If not specified, the program will not output this information.

`-o $offFile` Specifies a file to which write the final mesh in OFF format. If not specified, the program will write the mesh to standard output in OFF format.

`-t $tcFile` Specifies a file to which print the timing information. If not specified, the program will not output this information.

A.4.2 The `reconstruct_image` Program

SYNOPSIS

Table A.3: Optimality criteria

Optimality Criterion	Description	Reference
delaunay	(preferred-directions) Delaunay	[Section 3.2]
se	squared error (SE)	[Section 3.2]
abn	angle between normals (ABN)	[Section 3.2]
jnd	jump in normal derivatives (JND)	[Section 3.2]
el	Edge length (EL). Cost-based optimality criterion. For an edge e in the triangulation T, edgeCost(T,e) is defined as the length of e.	-
elabn	absolute mean curvature (AMC)	[Section 3.2]
eljnd	EL-weighted JND (ELJND)	[30, Section III.B]
dlp	deviations from linear polynomials (DLP)	[Section 3.2]
dp	distances from planes (DP)	[Section 3.2]
yms	Yu-Morse-Sederberg (YMS)	[29] and [30, Section III.B]
jndse	JND-weighted SE (JNDSE)	[Section 3.2]
elwse	EL-weighted SE (ELSE)	[Section 3.2]
sqse	shape-quality-weighted SE (SQSE)	[Section 3.2]
mase	minimum-angle-weighted SE (MASE)	[Section 3.2]
ghh	Garland-Heckbert hybrid (GHH)	[Section 3.2]

```
reconstruct_image [OPTIONS]
```

DESCRIPTION

This program reads a mesh in “.mesh” format from standard input (by default), and reconstructs an image from the mesh. The reconstructed image is saved in PGM format to a file specified by the option `-w`, and this option is required by the program. The program returns 0 for a normal exit and a nonzero value otherwise.

OPTIONS

`-w` `$recImage` Specifies the file to which write the reconstructed image in PGM format. This option is required.

`-o` `$offFile` Specifies the file that contains the OFF part of the “.mesh” file. If specified, the program will first read the non-OFF part of the “.mesh” file from standard input and then the OFF part from the specified file; otherwise, the program will read all parts of the “.mesh” file from standard input.

A.5 Examples of Software Usage

A few examples are provided in what follows to illustrate the usage of our software.

Example 1A. Suppose that we want to generate a mesh for the lena image and save the mesh along with other information that are printed to standard output to the file `lena+0.02.mesh`, with the following requirements:

- set the sampling density to be 2%;
- set the binomial filter order to be 3;
- set the sample-point-selection method to use the MMSODD as density function and the serpentine scan order in error diffusion for the ED method;
- set the mesh size tolerance to be 0;
- set the maximum visit count to be 15;
- set the program to skip the edge in the remainder of the currently running LOP if the maximum visit count has been exceeded for an edge;
- choose the optimality criterion used for the LOP during point insertion as ELSE; and

- use the LOP for final connectivity adjustment, with the SE criterion being applied.

The above task can be accomplished by invoking the command `generate_mesh` as follows:

```
generate_mesh -flena.pnm -D0.02 -O3 -Sywbs -z0 -Ielwse -c15 \
-v0 -m1 -Ese > lena+0.02.mesh
```

Example 1B. Suppose that we want to reconstruct an approximation of the lena image from the mesh generated in Example 1A and write the reconstructed image to `lena+rec.pgm`, this can be accomplished by running the command `reconstruct_image` as follows:

```
reconstruct_image -wlena+rec.pgm < lena+0.02.mesh
```

Example 2. Suppose that we want to generate a mesh for the bull image and save the mesh to an file `bull+mesh.off` in OFF format, with the following requirements:

- set the sampling density to be 1%;
- disable the binomial filter (i.e., set the filter order to be 1);
- set the sample-point-selection method to use the skewness as density function and the raster scan order in error diffusion for the ED method;
- set the mesh size tolerance to be 10;
- set the maximum visit count to be 20;
- set the program to terminate the currently running LOP if the maximum visit count for an edge has been exceeded;
- choose the optimality criterion used for the LOP during point insertion as MASE; and
- use the LLOP for final connectivity adjustment, with the SE criterion being applied.

The above task can be accomplished by invoking the command `generate_mesh` as follows:

```
generate_mesh -fbull.pnm -D0.01 -O1 -Ssked -z10 -Imase -l1 -c20 \
-v1 -m1 -Ese -obull+mesh.off
```

Example 3. Suppose that we want to generate a mesh for the cr image and print the mesh to standard output in OFF format, with the following requirements:

- set the sampling density to be 0.5%;
- set the order of the binomial filter to be 5;
- set the sample-point-selection method to use the MMSODD as density function and the raster scan order in error diffusion for the ED method;
- set the mesh size tolerance to be 0;
- set the maximum visit count to be 10;
- set the program to skip the edge in the remainder of the currently running LOP if the maximum visit count for an edge has been exceeded;
- choose the optimality criterion used for the LOP during point insertion as Delaunay;
- disable the final-connectivity-adjustment step;
- print the xyz-coordinates of the selected sample points to `cr+subset.dat`; and
- print the final triangulation to the iviewer-readable file `cr+tri.tri`;

The above task can be accomplished by invoking the command `generate_mesh` as follows:

```
generate_mesh -fcr.pnm -D0.005 -O5 -Sywb -z0 -Idelaunay -c10 \
-v0 -m0 -scr+subset.dat -icr+tri.tri
```

Bibliography

[1] S. A. Coleman, B. W. Scotney, and M. G. Herron. Image feature detection on content-based meshes. In *Proc. of IEEE International Conference on Image Processing*, volume 1, pages 844–847, 2002.

[2] M. Petrou, R. Piroddi, and A. Talebpour. Texture recognition from sparsely and irregularly sampled data. *Computer Vision and Image Understanding*, 102:95–104, 2006.

[3] M. Sarkis and K. Diepold. A fast solution to the approximation of 3-D scattered point data from stereo images using triangular meshes. In *Proc. of IEEE-RAS International Conference on Humanoid Robots*, pages 235–241, Pittsburgh, PA, USA, November 2007.

[4] J. G. Brankov, Y. Yang, and N. P. Galatsanos. Image restoration using content-adaptive mesh modeling. In *Proc. of IEEE International Conference on Image Processing*, volume 2, pages 997–1000, 2003.

[5] J. G. Brankov, Y. Yang, and M. N. Wernick. Tomographic image reconstruction based on a content-adaptive mesh model. *IEEE Trans. on Medical Imaging*, 23(2):202–212, February 2004.

[6] M. A. Garcia and B. X. Vintimilla. Acceleration of filtering and enhancement operations through geometric processing of gray-level images. In *Proc. of IEEE International Conference on Image Processing*, volume 1, pages 97–100, Vancouver, BC, Canada, 2000.

[7] D. Su and P. Willis. Demosaicing of colour images using pixel level data-dependent triangulation. In *Proc. of the Theory and Practice of Computer Graphics*, pages 16–23, 2003.

[8] D. Su and P. Willis. Image interpolation by pixel-level data-dependent triangulation. *Computer Graphics Forum*, 23(2):189–201, 2004.

[9] M. D. Adams. Progressive lossy-to-lossless coding of arbitrarily-sampled image data using the modified scattered data coding method. In *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 1017–1020, Taipei, Taiwan, April 2009.

[10] G. Ramponi and S. Carrato. An adaptive irregular sampling algorithm and its application to image coding. *Image and Vision Computing*, 19:451–460, 2001.

[11] P. Lechat, H. Sanson, and L. Labelle. Image approximation by minimization of a geometric distance applied to a 3D finite elements based model. In *Proc. of IEEE International Conference on Image Processing*, volume 2, pages 724–727, 1997.

[12] Y. Wang, O. Lee, and A. Vetro. Use of two-dimensional deformable mesh structures for video coding, part II–the analysis problem and a region-based coder employing an active mesh representation. *IEEE Trans. on Circuits and Systems for Video Technology*, 6(6):647–659, December 1996.

[13] F. Davoine, M. Antonini, J.-M. Chassery, and M. Barlaud. Fractal image compression based on Delaunay triangulation and vector quantization. *IEEE Trans. on Image Processing*, 5(2):338–346, February 1996.

[14] K.-L. Hung and C.-C. Chang. New irregular sampling coding method for transmitting images progressively. *IEE Proceedings Vision, Image and Signal Processing*, 150(1):44–50, February 2003.

[15] M. D. Adams. An efficient progressive coding method for arbitrarily-sampled image data. *IEEE Signal Processing Letters*, 15:629–632, 2008.

[16] I. Amidror. Scattered data interpolation methods for electronic imaging systems: a survey. *Journal of Electronic Imaging*, 11(2):157–176, April 2002.

[17] R. Franke and G. M. Nielson. Scattered data interpolation and applications: A tutorial and survey. In H. Hagen and D. Roller, editors, *Geometric Modeling: Methods and Applications*, pages 131–160. Springer-Lag, 1991.

[18] M. Garland and P. S. Heckbert. Fast polygonal approximation of terrains and height fields. Technical Report CMU-CS-95-181, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA, September 1995.

[19] Y. Yang, M. N. Wernick, and J. G. Brankov. A fast approach for accurate content-adaptive mesh generation. *IEEE Trans. on Image Processing*, 12(8):866–881, August 2003.

[20] M. D. Adams. A flexible content-adaptive mesh-generation strategy for image representation. *IEEE Trans. on Image Processing*, 20(9):2414–2427, September 2011.

[21] M. D. Adams. A highly-effective incremental/decremental Delaunay mesh-generation strategy for image representation. *Signal Processing*, 93(4):749–764, April 2013.

[22] X. Tu and M. D. Adams. Image representation using triangle meshes with explicit discontinuities. In *Proc. of IEEE Pacific Rim Conference on Communications, Computers and Signal Processing*, pages 97–101, Victoria, BC, Canada, August 2011.

[23] N. Dyn, D. Levin, and S. Rippa. Data dependent triangulations for piecewise liner interpolations. *IMA Journal of Numerical Analysis*, 10(1):137–154, 1990.

[24] N. Dyn, D. Levin, and S. Rippa. Algorithms for the construction of data dependent triangulations. In J. C. Mason and M. G. Cox, editors, *Algorithms for Approximation II*, pages 185–192. Chapman and Hall, London, 1990.

[25] E. Quak and L. L. Schumaker. Least squares fitting by linear splines on data dependent triangulations. In *Curves and surfaces*, pages 387–390. Academic Press Professional, Inc., 1991.

[26] S. Rippa. Adaptive approximation by piecewise linear polynomials on triangulations of subsets of scattered data. *SIAM Journal on Scientific and Statistical Computing*, 13(5):1123–1141, 1992.

[27] N. Dyn, D. Levin, and S. Rippa. Boundary correction for piecewise linear interpolation defined over data-dependent triangulations. *Journal of Computational and Applied Mathematics*, 39:179–192, 1992.

[28] N. Dyn. Data-dependent triangulations for scattered data interpolation and finite element approximation. *Applied Numerical Mathematics*, 12:89–105, 1993.

[29] X. Yu, B. S. Morse, and T. W. Sederberg. Image reconstruction using data-dependent triangulation. *IEEE Computer Graphics and Applications*, 21(3):62–68, May 2001.

[30] P. Li and M. D. Adams. A tuned mesh-generation strategy for image representation based on data-dependent triangulation. *IEEE Trans. on Image Processing*, 22(5):2004–2018, May 2013.

[31] S. Bougleux, G. Peyre, and L. D. Cohen. Image compression with anisotropic triangulations. In *Proc. of IEEE International Conference on Computer Vision*, pages 2343–2348, 2009.

[32] M. Garland and P. S. Heckbert. Fast triangular approximation of terrains and height fields. Technical report, 1997.

[33] J. L. Brown. Vertex based data dependent triangulations. *Computer Aided Geometric Design*, 8(3):239–251, 1991.

[34] L. Demaret and A. Iske. Scattered data coding in digital image compression. *Curve and Surface Fitting: Saint-Malo*, 2003:107–117, 2002.

[35] L. Demaret and A. Iske. Advances in digital image compression by adaptive thinning. *Annals of the MCFA*, 3:105–109, 2004.

[36] L. Demaret and A. Iske. Adaptive image approximation by linear splines over locally optimal Delaunay triangulations. *IEEE Signal Processing Letters*, 13(5):281–284, 2006.

[37] H.Weimer and J. Warren. Fast approximating triangulation of large scattered datasets. *Advances in Engineering software*, 30(6):389–400, 1999.

[38] L. L. Schumaker. Computing optimal triangulations using simulated annealing. *Computer Aided Geometric Design*, 10(3):329–345, 1993.

[39] B. Lehner, G. Umlauf, and B. Hamann. Image compression using data-dependent triangulations. In *Advances in Visual Computing*, pages 351–362. Springer, 2007.

[40] O. Kreylos and B. Hamann. On simulated annealing and the construction of linear spline approximations for scattered data. 7(1):17–31, 2001.

[41] S. Rippa. Long and thin triangles can be good for linear interpolation. *SIAM Journal on Numerical Analysis*, 29(1):257–270, 1992.

[42] N. Dyn, M. S. Floater, and A. Iske. Adaptive thinning for bivariate scattered data. *Journal of Computational and Applied Mathematics*, 145(2):505–517, 2002.

[43] B. Delaunay. Sur la sphere vide. *Bulletin of the Academy of Sciences of the USSR, Classe des Sciences Mathematiques et Naturelle*, 7(6):793–800, 1934.

[44] J. Shewchuk. What is a good linear finite element? interpolation, conditioning, anisotropy, and quality measures. *University of California at Berkeley*, 73, 2002.

[45] C. Dyken and M. S. Floater. Preferred directions for resolving the non-uniqueness of Delaunay triangulations. *Computational Geometry—Theory and Applications*, 34:96–101, 2006.

[46] C. L. Lawson. Software for C^1 surface interpolation. In J. R. Rice, editor, *Mathematical Software III*, pages 161–194. Academic Press, New York, NY, USA, 1977.

[47] K. Wang, C.-P. Lo, G. A. Brook, and H. R. Arabnia. Comparison of existing triangulation methods for regularly and irregularly spaced height fields. *International Journal of Geographical Information Science*, 15(8):743–762, 2001.

[48] M. Bertram, J. C. Barnes, B. Hamann, K. I. Joy, H. Pottman, and D. Wushour. Piecewise optimal triangulation for the approximation of scattered data in the plane. *Computer Aided Geometric Design*, 17(8):767–787, 2000.

[49] X. Ma and M. D. Adams. An improved method for generating triangle-mesh models of images. Halifax, NS, Canada, August 2014. Available online at `https://projects.cs.dal.ca/cccg2014/proceedings/papers/paper10.pdf`.

[50] X. Ma and M. D. Adams. An improved error-diffusion approach for generating mesh models of images. Submitted to *Signal Processing* on October 29, 2014.

[51] M. Aubury and W. Luk. Binomial filters. *Journal of VLSI Signal Processing*, 12(1):35–50, 1996.

[52] R. A. Haddad and A. N. Akansu. A class of fast gaussian binomial filters for speech and image processing. *IEEE Trans. on Signal Processing*, 39(3):723–727, 1991.

[53] K. Fleischer and D. Salesin. Accurate polygon scan conversion using half-open intervals. In *Graphics Gems III*, pages 362–365, 1992.

[54] P. K. Agarwal and S.Suri. Surface approximation and geometric partitions. *Proc. of ACM-SIAM Symposium on Discrete Algorithms*, pages 24–33, January 1994.

[55] R. W. Floyd and L. Steinberg. An adaptive algorithm for spatial greyscale. *Proc. of the Society for Information Display*, 17(2):75–77, 1976.

[56] C. L. Lawson. Transforming triangulations. In *Discrete Math.*, volume 3, pages 365–372, 1972.

[57] L. Alboul, G. Kloosterman, C. Traas, and R van Damme. Best data-dependent triangulations. *Journal of Computational and Applied Mathematics*, 119:1–12, 2000.

[58] JPEG-2000 test images. ISO/IEC JTC 1/SC 29/WG 1N 545, July 1997.

[59] USC-SIPI image database. http://sipi.usc.edu/database, 2014.

[60] Kodak lossless true color image suite. http://r0k.us/graphics/kodak, 2014.

[61] CGAL, Computational Geometry Algorithms Library. http://www.cgal.org.

[62] M. D. Adams. SPL, Signal Processing Library, 2014. SPL manual page: http://www.ece.uvic.ca/~frodo/SPL/manual/current/html.

[63] Boost C++ Library. http://www.boost.org.

[64] pgm, netpbm grayscale image format, 2014. http://netpbm.sourceforge.net/doc/pgm.html.

[65] Off, object file format, 2014. http://segeval.cs.princeton.edu/public/off_format.html.