A Tuned Mesh-Generation Strategy for Image Representation Based on Data-Dependent Triangulation

Ping Li and Michael D. Adams, *Senior Member, IEEE*

Abstract

A mesh-generation framework for image representation based on data-dependent triangulation (DDT) is proposed. The proposed framework is a modified version of the frameworks of Rippa and Garland and Heckbert that facilitates the development of more effective mesh-generation methods. As the proposed framework has several free parameters, the effects of different choices of these parameters on mesh quality are studied, leading to the recommendation of a particular set of choices for these parameters. A mesh-generation method is then introduced that employs the proposed framework with these best parameter choices. This method is demonstrated to produce meshes of higher quality (both in terms of squared error and subjectively) than those generated by several competing approaches, at a relatively modest computational and memory cost.

Index Terms

Image representation, nonuniform sampling, triangle mesh, data-dependent triangulation, mesh generation.

Manuscript received ...; revised ...; accepted .... Date of publication ...; date of current version .... This work was supported by the Natural Sciences and Engineering Research Council of Canada. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Bulent Sankur.

P. Li is with Qualcomm Canada Inc., Markham, ON, Canada, L3T 7W3 (e-mail: pingli@qti.qualcomm.com).

M. D. Adams is with the Department of Electrical and Computer Engineering, University of Victoria, Victoria, BC, V8W 3P6, Canada (e-mail: mdadams@ece.uvic.ca).

Digital Object Identifier ...

I. INTRODUCTION

Traditionally, most commonly-used image representations are based on uniform (lattice-based) sampling. Because most images are nonstationary, however, uniform sampling is usually far from being optimal. When uniform sampling is employed, the sampling density will inevitably be too high in regions where the image is changing slowly and too low in regions where the image is changing rapidly. For this reason, there has been a growing interest in image representations that utilize nonuniform sampling, where the sampling density is adaptive to the image content. Furthermore, images also possess geometric structure, which is inherent in their edges (i.e., discontinuities). In many cases, image representations based on nonuniform sampling can more easily capture this geometric structure. By using a more compact image representation afforded by nonuniform sampling, we can often reduce computational complexity or obtain improved results in the application at hand. For example, image representations based on nonuniform sampling have proven beneficial for such tasks as feature detection [1], pattern recognition [2], computer vision [3], restoration [4], tomographic reconstruction [5], filtering [6], interpolation [7], [8], and image/video coding [9]–[15].

Many classes of image representations based on nonuniform sampling have been proposed to date, such as inverse distance weighted methods [16], [17], radial basis function methods [16], [17], Voronoi and natural-neighbour interpolation methods [16], and finite-element methods [16], [17] including triangle meshes. For a good summary of all of these approaches, the reader is referred to the excellent survey papers [16] and [17]. Of the above classes of approaches, the one based on triangle meshes has become quite popular. With a mesh model of an image, the image domain is partitioned by a triangulation into a set of (triangle) faces, and then over each face of the triangulation an approximating function is constructed. One popular subclass of mesh representations is those based on Delaunay triangulations [18]. To whatever extent is possible, Delaunay triangulations avoid long thin (i.e., sliver) triangles which can lead to very poor approximations *if not well chosen*. In the Delaunay case, the connectivity of the triangulation (i.e., how the points in the triangulation are connected by edges) is determined solely by the geometry (i.e., position) of the points being triangulated. Another subclass of mesh representations is those based on data-dependent triangulations (DDTs). It is this particular subclass that is of interest herein. In the case of DDTs, the connectivity of the triangulation is chosen in a way that depends on the data set from which the points to be triangulated originated (and not just the geometry of those points). Consequently, mesh generation consists of two distinct (but related) subproblems: 1) the selection of the sample points (i.e., the vertices of triangulation), and 2) the selection of the connectivity of the triangulation. Since, unlike the Delaunay case, DDTs can have their connectivity chosen arbitrarily, DDTs offer vastly greater

flexibility, and theoretically have the potential to perform much better than their Delaunay counterparts *if well chosen* [19]. In practice, however, due to this increased flexibility, it is much more difficult to develop highly-effective *computationally-efficient* mesh-generation schemes that are based on DDTs, as compared to the Delaunay case.

To date, numerous DDT-based mesh-generation methods have been proposed [7], [8], [20]–[34]. Unfortunately, many of these methods (e.g., [7], [8], [23]–[32]) concern themselves with only the problem of triangulation-connectivity selection (i.e., the second subproblem mentioned above). That is, they assume that the sample points are given, or have already been chosen through some unspecified means, and then only concern themselves with selecting the connectivity of the triangulation. Such an assumption is not very realistic in many applications. For this reason, methods that select both the sample points and triangulation connectivity are of great practical interest. Another further difficulty with DDT-based mesh-generation methods is that they can often have very high computational cost. For example, some DDT-based schemes [32]–[34] are based on simulated annealing, which is very computationally expensive. Another DDT-based method proposed in [28] takes several iterations and requires between 0.5 and 5 seconds per iteration for an 80×80 image. Furthermore, this method only considers triangulation connectivity. Therefore, as one can well imagine, in the case of the more difficult problem of choosing both the sample points and triangulation connectivity, mesh-generation methods can potentially become very computationally complex.

Of those schemes that are relatively fast and choose both the sample points and triangulation connectivity, a particularly good one was proposed by Garland and Heckbert [20, Algorithm IV] (with the quality threshold parameter `qthresh` chosen as 0.5 and an L^2 error measure), which we henceforth refer to as the Garland-Heckbert (GH) method. The GH method is associated with a basic framework for mesh generation, which is very similar to a framework proposed even earlier by Rippa [22]. In this paper, we build on the aforementioned past work of Rippa and Garland and Heckbert. We propose a modified version of these earlier frameworks that facilitates the development of better performing mesh-generation methods, and then we introduce a highly-effective mesh-generation method based on our framework. As we shall see, our proposed mesh-generation method produces mesh representations of images with lower approximation error than those generated by other competing schemes. At the same time, the computational and memory costs of our proposed method are relatively modest.

The remainder of this paper is organized as follows. Section II introduces some background information on meshes for image representation. In Section III, our proposed mesh-generation framework is presented, which has several free parameters that must be chosen in order to produce a fully-specified mesh-

generation method. Section IV examines which choices for these free parameters are most effective, leading us to propose a specific mesh-generation method. Then, in Section V, the performance of our proposed mesh-generation method is compared to that of several competing schemes, with our method proving to be superior. Finally, Section VI concludes the paper with a summary of our key results.

II. Mesh Models of Images

Before proceeding further, a brief digression is in order to introduce some notation and basic terminology used herein. For a set S, $|S|$ denotes the cardinality of S. The symbol $\|\cdot\|$ denotes the 2-norm. Formally, a triangulation of a set V of points is a set $T = \{f_i\}_{i=0}^{|T|-1}$ of nondegenerate open triangles satisfying the following conditions: 1) the set of all vertices of triangles in T is V; 2) every edge of a triangle in T contains only two points from V; 3) $\cup_{i=0}^{|T|-1} \overline{f_i}$ is the convex hull of V; and 4) $f_i \cap f_j = \emptyset$ for $i \neq j$. For a triangulation T, the vertices, edges, and faces of T are denoted as $\boldsymbol{\mathcal{V}}(T)$, $\boldsymbol{\mathcal{E}}(T)$, and $\boldsymbol{\mathcal{F}}(T)$, respectively.

Consider an integer-valued image function ϕ defined on $\Lambda = \{0, 1, \ldots, W-1\} \times \{0, 1, \ldots, H-1\}$ (i.e., a rectangular grid of width W and height H). With a mesh model of an image, the image domain is partitioned by a triangulation into a set of (triangle) faces and then over each face of the triangulation an approximating function is constructed. A mesh model is completely characterized by a triangulation T covering the image domain Λ as well as the values of ϕ for each point $p \in \boldsymbol{\mathcal{V}}(T)$. We refer to each element of $\boldsymbol{\mathcal{V}}(T)$ as a sample point. Given T, a function $\hat{\phi}_T$ that interpolates ϕ at the points in $\boldsymbol{\mathcal{V}}(T)$ is constructed as follows. First, we form a continuous piecewise-linear function $\tilde{\phi}_T$. For each (triangle) face $f \in \boldsymbol{\mathcal{F}}(T)$, $\tilde{\phi}_T$ is chosen as the unique linear function that interpolates ϕ at the three vertices of f. To ensure that $\hat{\phi}_T$ is integer valued (just like ϕ), we choose $\hat{\phi}_T$ as $\hat{\phi}_T(p) = \mathrm{round}(\tilde{\phi}_T(p))$ for all $p \in \Lambda$, where round denotes an operator that rounds to an integer value. The set $\boldsymbol{\mathcal{V}}(T)$ must always include the extreme convex-hull points of Λ (i.e., the four corners of the image bounding box) so that the triangulation of $\boldsymbol{\mathcal{V}}(T)$ covers all points in Λ. As a matter of terminology, the size and sampling density of the mesh model T are defined as $|\boldsymbol{\mathcal{V}}(T)|$ (i.e., the number of vertices in T) and $|\boldsymbol{\mathcal{V}}(T)| \,/\, |\Lambda|$, respectively.

The mesh-generation problem that we address in this paper can be succinctly stated as follows: Given ϕ and a desired number N of sample points, find the mesh model T of ϕ with $|\boldsymbol{\mathcal{V}}(T)| = N$ that minimizes the measure ϵ_T of the difference between ϕ and the approximation $\hat{\phi}_T$. In our work, the mean squared error (MSE) is used as the error measure, so that

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

Fig. 1. Edge flip example. Part of (a) a triangulation with an edge $\overline{v_i v_j}$ and (b) the new triangulation obtained after the edge $\overline{v_i v_j}$ is transformed to the edge $\overline{v_k v_l}$ by an edge flip.

Herein, the MSE is typically expressed in terms of the peak signal-to-noise ratio (PSNR), which is defined as $\mathsf{PSNR} = 20 \log_{10} \left(\frac{2^\rho - 1}{\sqrt{\epsilon}} \right)$, where ρ is the number of bits/sample in the image ϕ. Finding good *computationally-efficient* methods for solving the above-stated mesh-generation problem is quite challenging, since problems like this are known to be NP hard [35].

III. Proposed Mesh-Generation Framework

The mesh-generation framework proposed herein was inspired by the frameworks of Rippa [22] (known by the name "COMPRESS" therein) and Garland and Heckbert [20] (known as "Algorithm IV" therein). Both the Rippa and Garland-Heckbert frameworks employ the well-known local optimization procedure (LOP) [30] of Lawson. Our proposed framework also utilizes a variant of the LOP. Since knowledge of the LOP is essential to the understanding of our framework, we will first describe the LOP below.

To begin, we introduce a few basic definitions and facts about triangulations. An edge e in a triangulation is said to be flippable if it has two incident faces (i.e., is not on the triangulation boundary) and the union of these two faces is a *strictly convex* quadrilateral q. If e is flippable, a valid triangulation is obtained if e is deleted from the triangulation and replaced by the other diagonal of the quadrilateral q. This transformation is known as an edge flip. For example, in Fig. 1, the edge $\overline{v_i v_j}$ in Fig. 1(a) is transformed to the edge $\overline{v_k v_l}$ in Fig. 1(b) by an edge flip. Any triangulation of a set of points can be obtained from any other triangulation of the same set of points by a finite sequence of edge flips [36], [37]. Moreover, since an edge flip does not change the number of edges in a triangulation, this further implies that every triangulation of a set of points has the same number of edges.

The fact that every triangulation is reachable from every other triangulation via edge flips (as described above) motivated Lawson to propose the so called LOP [30], an algorithm for finding an optimal triangulation of a set of points via edge flips. To cast the triangulation problem as an optimization, we define a rule, called an edge-flip criterion, that determines, for a flippable edge e, if the triangulation

with the edge e is preferred over the triangulation obtained if e were transformed to e' by an edge flip. The edge-flip criterion is specified as a binary-valued decision function, denoted $\mathrm{isPreferred}$, where $\mathrm{isPreferred}(e)$ is one if e is preferred to e', and zero otherwise. As a matter of terminology, an edge e is said to be optimal if: 1) it is not flippable; or 2) it is flippable and $\mathrm{isPreferred}(e) = 1$. A triangulation is said to be (locally) optimal if each one of its edges is optimal. An edge whose optimality is uncertain is said to be suspect. Note that, by definition, an edge that is not flippable cannot be suspect. In short, the LOP tests any suspect edges for optimality, and performs edge flips to eliminate any suspect edge that is not optimal. Letting S denote the current set of suspect edges, the variant of the LOP used herein consists of the following steps:

1) Initialize S to contain all suspect edges (i.e., edges that are flippable but whose optimality has not yet been tested).
2) If $|S| = 0$ (i.e., S is empty), then stop.
3) Remove an edge e from S.
4) If e is not flippable or e has been visited more than 5 times since the LOP started, go to step 2.
5) Let q denote the (strictly convex) quadrilateral formed by the union of the two faces incident on e, and let e' denote the edge obtained by flipping e (i.e., the other diagonal of q). If $\mathrm{isPreferred}(e) = 0$ (i.e., e is not optimal according to whatever edge-flip criterion is in effect), apply an edge flip to e (to produce e'), and add to S any newly suspect edges resulting from the edge flip. Which edges become newly suspect as a result of the edge flip depend on the specific choice of $\mathrm{isPreferred}$, and will be addressed in more detail later in Section III-B, after we have introduced the various edge-flip criteria considered herein. In practice, however, these edges are in a small neighbourhood about e.
6) Go to step 2.

In passing, we note that the LOP is only guaranteed to produce a locally (as opposed to globally) optimal triangulation, and the locally optimal triangulation produced depends on the order in which edges are flipped. The algorithm presented above differs slightly from the LOP as proposed by Lawson. In particular, a limit is placed on the number of times that an edge can be tested for optimality during the LOP process via the second condition appearing in step 4. This extra condition is necessary in order to ensure that the algorithm does not become trapped in a cycle, repeating the same sequence of edge flips indefinitely. Cycles can arise for two reasons. First, due to the effects of finite-precision arithmetic (i.e., roundoff error), decisions regarding the optimality of an edge can occasionally be made in an inconsistent manner, leading to cycles. Second, in our work, we allow for the use of more general edge-flip criteria that, even

in the absence of roundoff error, can occasionally result in cycles (i.e., the GHH, SQSE, and JNDSE criteria introduced later in Section III-B). In the case of our software implementation, which is robust to roundoff error, cycles can only occur for the second reason.

Having introduced the (slightly modified) LOP used in our work, we can now present our proposed mesh-generation framework. For the remainder of this paper, including this section, the quantities Λ, ϕ, $\tilde{\phi}$, and $\hat{\phi}$ are as defined previously in Section II. With our framework, for a given triangulation T, each point in the image domain Λ is assigned to *exactly one* face in T. If a point is strictly inside a face, the point is assigned to that face. If a point is on an edge or is a vertex in T, a scheme similar to [38] is used to uniquely assign the point to a face. The set of all points in Λ belonging to the face f in T is denoted $\boldsymbol{\mathcal{P}}_T(f)$.

As input, our framework takes an image ϕ (defined on Λ) and a target number N of sample points for the mesh to be generated. Our proposed framework is iterative in nature. It starts with a nearly empty mesh and adds points to the mesh until the desired sampling density is achieved. Let T denote the triangulation in the current iteration. Our framework then consists of the following steps (in order):

1) *Initial triangulation.* Initially choose the triangulation T as a triangulation of the extreme convex-hull points of the image domain Λ (i.e., the four corner points of the image bounding box).
2) *Initial connectivity adjustment.* Adjust the connectivity of the triangulation by applying the LOP (described earlier) choosing the edge-flip criterion $\mathrm{isPreferred}$ as $\mathrm{isPreferred} = \mathrm{isPreferred}_{\mathsf{main}}$, where $\mathrm{isPreferred}_{\mathsf{main}}$ is a free parameter of our framework. Initially, when the LOP is invoked, all flippable edges in T are marked as suspect.
3) If the target number of sample points has been reached (i.e., $|\boldsymbol{\mathcal{V}}(T)| \geq N$), go to step 8.
4) *Point selection.* Select a new point $p^* \in \Lambda \backslash \boldsymbol{\mathcal{V}}(T)$ to add to the triangulation T. This is accomplished in two steps. First, select a face f^* in T into which a new point is to be inserted, as given by

$$f^* = \mathrm{selFace},$$

where $\mathrm{selFace}$ is a function (implicitly depending on T) that embodies the face-selection process and is a free parameter of our framework. Second, having chosen the face f^*, select a candidate point p^* belonging to f^* and not currently in T (i.e., $p^* \in \boldsymbol{\mathcal{P}}_T(f^*) \setminus \boldsymbol{\mathcal{V}}(T)$) for insertion, as given by

$$p^* = \mathrm{selCand}(f^*), \tag{2}$$

where $\mathrm{selCand}$ is a function that embodies the candidate-selection process and is a free parameter of our framework.

Fig. 2. Point insertion example. Part of a triangulation showing how the new vertex p^* is inserted (a) inside a triangle $v_iv_jv_k$ and (b) on an edge v_iv_k.

5) *Point insertion.* Insert p^* into the triangulation T. If p^* is strictly inside a face in T, say face $v_iv_jv_k$, we connect p^* by new edges to each vertex of its containing face, as shown in Fig. 2(a). If p^* is on an edge in T, say on edge $\overline{v_iv_k}$, we split this edge at p^* and, for each face f incident on $\overline{v_iv_k}$, we connect p^* with a new edge to the vertex in f that is opposite the edge $\overline{v_iv_k}$, as shown in Fig. 2(b).
6) *Main connectivity adjustment.* Adjust the connectivity of the triangulation by applying the LOP, with the edge-flip criterion $\mathrm{isPreferred}$ chosen as $\mathrm{isPreferred} = \mathrm{isPreferred}_{\mathsf{main}}$ and the suspect edges initially chosen as all edges whose optimality could have been changed by the insertion of p^* in the previous step (i.e., step 5). The edges whose optimality can be affected by the insertion of p^* depend on the particular choice of $\mathrm{isPreferred}_{\mathsf{main}}$, and will be discussed in more detail later in Section III-B, after we have introduced the various edge-flip criteria considered herein. In practice, these edges are in a relatively small neighbourhood about p^*.
7) Go to step 3.
8) *Final connectivity adjustment (optional).* If the (optional) final connectivity-adjustment step is enabled, continue; otherwise, stop. Adjust the connectivity of the triangulation by applying the LOP, with the edge-flip criterion $\mathrm{isPreferred}$ chosen as $\mathrm{isPreferred} = \mathrm{isPreferred}_{\mathsf{final}}$, where $\mathrm{isPreferred}_{\mathsf{final}}$ is a free parameter of our framework. Initially, when the LOP is invoked, all flippable edges in the triangulation are marked as suspect.

From above, one can see that our framework requires the choice of several free parameters in order to arrive at a completely-specified method for mesh generation. In particular, we must choose: 1) a point-selection strategy that consists of a face-selection policy $\mathrm{selFace}$ together with a candidate-selection policy $\mathrm{selCand}$; 2) an edge-flip criterion $\mathrm{isPreferred}_{\mathsf{main}}$ to be used in the initial and main connectivity adjustment, called the main edge-flip criterion; 3) a choice of whether to perform the optional final connectivity adjustment; 4) if final connectivity adjustment is to be performed, an edge-flip criterion

$\mathrm{isPreferred}_{\mathsf{final}}$ to be used for this purpose, called the final edge-flip criterion. Note that it is the intention of our framework that $\mathrm{isPreferred}_{\mathsf{main}}$ and $\mathrm{isPreferred}_{\mathsf{final}}$ be chosen differently. As for how the above parameters might be chosen, we defer this discussion until later.

Since, as mentioned above, our proposed framework is derived from the frameworks of Rippa [22] and Garland and Heckbert [20], it is beneficial to briefly outline the main differences between our framework and the ones from which it is derived. The most important difference is that neither the Rippa nor Garland-Heckbert framework has an equivalent to the final connectivity-adjustment step (i.e., step 8) in our framework. As we shall see later, final connectivity adjustment plays a crucial role in allowing highly effective mesh-generation methods to be synthesized. Another difference is that, in the Rippa and Garland-Heckbert frameworks, the point-selection process (corresponding to step 4 above) is not *logically* viewed as being split into two smaller steps, with the first choosing a face and the second choosing a point within the face. Both approaches always choose p^* as the point $p \in \Lambda \setminus \boldsymbol{\mathcal{V}}(T)$ for which $\left|\hat{\phi}_T(p) - \phi(p)\right|$ is greatest. That is, the Rippa and Garland-Heckbert frameworks essentially replace (2) in our framework with the fixed choice

$$p^* = \operatorname*{arg\,max}_{p \in \Lambda \setminus \boldsymbol{\mathcal{V}}(T)} \left|\hat{\phi}_T(p) - \phi(p)\right|, \tag{3}$$

in which case there is no explicit face-selection process ($\mathrm{selFace}$) per se. As we shall see later, the point-selection scheme given by (3) leaves much to be desired.

In passing, we note that mesh-generation methods derived from our proposed framework (as well as the Rippa and Garland-Heckbert frameworks) have a number of desirable characteristics. In particular, because our framework is based strictly on the refinement of an initial coarse mesh (with 4 vertices), the current mesh size never exceeds the target mesh size (i.e., N vertices) at any point during mesh generation. This is important in order to minimize memory usage (and often computational complexity as well). In contrast, frameworks/methods based on mesh simplification typically have a much greater peak mesh size. Another desirable characteristic of our proposed framework is that it can easily accommodate stopping criteria for the mesh-generation process that are based on either sampling density or a prescribed error tolerance (in step 3).

As seen above, our proposed framework requires the choice of several free parameters in order to arrive at a completely-specified method for mesh generation. For example, we must specify face-selection and candidate-selection policies as well as various edge-flip criteria. So far, we have not made any suggestions as to how these items might be chosen. In what follows, we introduce a variety of possible choices for these items as considered in our work.

A. Face- and Candidate-Selection Policies

In step 4 of our framework (i.e., point selection), we must choose a face f^* in which to select a new point for insertion. This face-selection policy is embodied by the function $\mathrm{selFace}$. In our work, we considered two face-selection policies. The first policy, called greatest absolute error (GAE), chooses $\mathrm{selFace}$ as

$$\mathrm{selFace}_{\mathsf{GAE}} = \operatorname*{arg\,max}_{f \in U(T)} \max_{p \in \boldsymbol{\mathcal{P}}_T(f)} \left| \hat{\phi}_T(p) - \phi(p) \right|,$$

where $U(T)$ is all faces f in $\boldsymbol{\mathcal{F}}(T)$ such that $\boldsymbol{\mathcal{P}}_T(f) \neq \emptyset$. That is, $\mathrm{selFace}$ is defined to choose the face in T that contains the point where the original image and approximation differ most. The second policy, called greatest squared error (GSE), selects $\mathrm{selFace}$ as

$$\mathrm{selFace}_{\mathsf{GSE}} = \operatorname*{arg\,max}_{f \in U(T)} \sum_{p \in \boldsymbol{\mathcal{P}}_T(f)} \left(\hat{\phi}_T(p) - \phi(p) \right)^2,$$

where $U(T)$ is all faces f in $\boldsymbol{\mathcal{F}}(T)$ such that $\boldsymbol{\mathcal{P}}_T(f) \neq \emptyset$. That is, $\mathrm{selFace}$ is defined to choose the face in T with the largest squared error.

In step 4 of our framework (i.e., point selection), once a face f^* has been chosen, we must select a candidate point p^* within that face for insertion. This candidate-selection policy is embodied by the function $\mathrm{selCand}$. In our work, we considered three candidate-selection policies. The first policy, called peak absolute error (PAE), selects $\mathrm{selCand}$ as

$$\mathrm{selCand}_{\mathsf{PAE}}(f) = \operatorname*{arg\,max}_{p \in \boldsymbol{\mathcal{P}}_T(f) \setminus \boldsymbol{\mathcal{V}}(T)} \left| \hat{\phi}_T(p) - \phi(p) \right|.$$

That is, of all candidate points in the face, this policy selects the point at which the absolute error is greatest.

The second policy, which is newly proposed herein and called approximate minimum squared error (AMSE), chooses p^* in two steps. First, we choose a set Ω of test points to consider as candidates for insertion. We then choose p^* from Ω. If $|\boldsymbol{\mathcal{P}}_T(f^*) \setminus \boldsymbol{\mathcal{V}}(T)| > 8$, Ω is chosen as the 8 points p in $\boldsymbol{\mathcal{P}}_T(f^*) \setminus \boldsymbol{\mathcal{V}}(T)$ for which $\left| \hat{\phi}_T(p) - \phi(p) \right|$ is greatest; otherwise, we choose $\Omega = \boldsymbol{\mathcal{P}}_T(f^*) \setminus \boldsymbol{\mathcal{V}}(T)$. (As an aside, we note that the value of 8 here was chosen based on experimentation.) Given Ω, we then choose $\mathrm{selCand}$ as

$$\mathrm{selCand}_{\mathsf{AMSE}}(f) = \operatorname*{arg\,min}_{t \in \Omega} \sum_{p \in \boldsymbol{\mathcal{P}}_T(f)} \left(\hat{\phi}_{\Upsilon(t)}(p) - \phi(p) \right)^2,$$

where $\Upsilon(t)$ denotes the triangulation that would be obtained if the point t were inserted into T. That is, $\mathrm{selCand}$ is defined to select the point in Ω whose insertion would result in the least squared error over the face f^*.

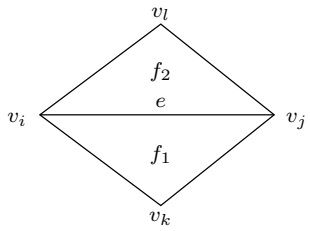


Fig. 3. An edge $e = \overline{v_i v_j}$ that is a diagonal of the quadrilateral $v_i v_k v_j v_l$ in the triangulation T.

The third policy (also newly proposed), called hybrid, simply employs the PAE policy until the sampling density of the mesh reaches 25% of the desired sampling density, with the AMSE policy being used thereafter. This policy is motivated largely by the desire to save computation. By using the less-computationally costly PAE policy initially, computational cost can be significantly reduced (relative to the AMSE policy).

Lastly, we note that choosing the face-selection and candidate-selection policies as GAE and PAE, respectively, is mathematically equivalent to the point-selection scheme used in the Rippa and Garland-Heckbert frameworks, given earlier by (3).

B. Edge-Flip Criteria

In our framework, steps 2, 6, and 8 each employ the LOP, and the LOP requires the specification of an edge-flip criterion. So far, we have not commented on how this edge-flip criterion might be chosen. In what follows, we introduce twelve edge-flip criteria that were considered in our work. Herein, we employ two different classes of edge-flip criterion. Both are based on the idea of assigning costs to edges, and then making the decision of whether to flip an edge based on these costs. The main difference between these two classes is in how they use these edge-cost functions in order to make decisions.

Before we can introduce any of the edge-flip criteria, we must first introduce the edge-cost functions that these criteria employ. Each edge-cost function assigns a cost to an edge e in the triangulation T. Let $v_n = (x_n, y_n)$ denote the nth vertex in T. Furthermore, let $e = \overline{v_i v_j}$. In what follows, in the case that e is not a boundary edge (and must therefore have two incident faces), its two incident faces are denoted as f_1 and f_2, where f_1 is triangle $v_i v_k v_j$ and f_2 is triangle $v_i v_j v_l$. The preceding definitions are illustrated in Fig. 3. Furthermore, let P_1 and P_2 denote the linear (i.e., planar) functions obtained when the approximating function $\tilde{\phi}_T$ has its domain restricted to f_1 and f_2, respectively, where $P_1(x, y) = a_1 x + b_1 y + c_1$, $P_2(x, y) = a_2 x + b_2 y + c_2$, and a_1, b_1, c_1, a_2, b_2, and c_2 are real constants. Lastly, we introduce a few additional definitions needed in what follows. The approximate diameter of the face

f, denoted $\mathrm{diam}(f)$, is defined as the length of the longest side of the (smallest axis-aligned) bounding box of the face f. The shape quality of a face f, denoted $\mathrm{sq}(f)$, is defined as $\mathrm{sq}(f) = \mathrm{area}(f)/\,\mathrm{diam}(f)$, where $\mathrm{area}(f)$ denotes the area of the face f.

The first class of edge-flip criterion is associated with edge-cost functions that assign a cost to *every* edge in the triangulation T. This class makes use of the following seven edge-cost functions: 1) $\mathrm{edgeCost}_{\mathsf{ABN}}$, the angle between normals (ABN) from [23]; 2) $\mathrm{edgeCost}_{\mathsf{JND}}$, the jump in normal derivatives (JND) from [23]; 3) $\mathrm{edgeCost}_{\mathsf{DLP}}$, the deviations from linear polynomials (DLP) from [23]; 4) $\mathrm{edgeCost}_{\mathsf{DP}}$, the distances from planes (DP) from [23]; 5) $\mathrm{edgeCost}_{\mathsf{YMS}}$, the Yu-Morse-Sederberg (YMS) cost from [28]; 6) $\mathrm{edgeCost}_{\mathsf{ELABN}}$, the edge-length-weighted ABN (ELABN) from [26], [27]; and 7) $\mathrm{edgeCost}_{\mathsf{ELJND}}$, the edge-length-weighted JND (ELJND) which is newly proposed herein. The definitions of these functions are as follows. For a nonboundary edge e (which must have two incident faces) in the triangulation T, we define

$$\mathrm{edgeCost}_{\mathsf{ABN}}(T,e) = \arccos\left[\frac{(a_1,b_1,-1)\cdot(a_2,b_2,-1)}{\|(a_1,b_1,-1)\|\,\|(a_2,b_2,-1)\|}\right], \tag{4a}$$

$$\mathrm{edgeCost}_{\mathsf{JND}}(T,e) = \left|(n_{\mathsf{x}},n_{\mathsf{y}})\cdot[(a_1,b_1)-(a_2,b_2)]\right|, \tag{4b}$$

$$\mathrm{edgeCost}_{\mathsf{DLP}}(T,e) = \left\|\left(\left|P_1(x_l,y_l)-\phi(x_l,y_l)\right|,\left|P_2(x_k,y_k)-\phi(x_k,y_k)\right|\right)\right\|, \tag{4c}$$

$$\mathrm{edgeCost}_{\mathsf{DP}}(T,e) = \left\|\left(\mathrm{dist}(P_1,(x_l,y_l,\phi(x_l,y_l))),\mathrm{dist}(P_2,(x_k,y_k,\phi(x_k,y_k)))\right)\right\|, \tag{4d}$$

$$\mathrm{edgeCost}_{\mathsf{ELABN}}(T,e) = \|v_i-v_j\|\,\mathrm{edgeCost}_{\mathsf{ABN}}(T,e), \tag{4e}$$

$$\mathrm{edgeCost}_{\mathsf{YMS}}(T,e) = \|(a_1,b_1)\|\,\|(a_2,b_2)\| - (a_1,b_1)\cdot(a_2,b_2), \quad \text{and} \tag{4f}$$

$$\mathrm{edgeCost}_{\mathsf{ELJND}}(T,e) = \|v_i-v_j\|\,\mathrm{edgeCost}_{\mathsf{JND}}(T,e), \tag{4g}$$

where $(n_{\mathsf{x}},n_{\mathsf{y}})$ is a unit vector normal to e and $\mathrm{dist}(P_\alpha,(x,y,z)) = \frac{|P_\alpha(x,y)-z|}{\|(a_\alpha,b_\alpha,-1)\|}$ (and P_i, a_i, and b_i are as defined in the previous paragraph). Note that $(a_i,b_i,-1)$ is a normal to the plane defined by P_i; (a_i,b_i) is the gradient of P_i; and $\mathrm{dist}(P_i,(x,y,z))$ is the shortest distance from the point (x,y,z) to the plane defined by P_i. For a boundary edge e (which has only one incident face), we simply define $\mathrm{edgeCost}_{\mathsf{ABN}}(T,e) = 0$, $\mathrm{edgeCost}_{\mathsf{JND}}(T,e) = 0$, $\mathrm{edgeCost}_{\mathsf{DLP}}(T,e) = 0$, $\mathrm{edgeCost}_{\mathsf{DP}}(T,e) = 0$, $\mathrm{edgeCost}_{\mathsf{YMS}}(T,e) = 0$, $\mathrm{edgeCost}_{\mathsf{ELABN}}(T,e) = 0$, and $\mathrm{edgeCost}_{\mathsf{ELJND}}(T,e) = 0$.

The second class of edge-flip criterion is associated with edge-cost functions that assign a cost to every *flippable* edge in the triangulation. This class makes use of the following five edge-cost functions: 1) $\mathrm{edgeCost}_{\mathsf{D}}$, the (preferred-directions) Delaunay (D) cost [39]; 2) $\mathrm{edgeCost}_{\mathsf{SE}}$, the squared error (SE) from [25]; 3) $\mathrm{edgeCost}_{\mathsf{GHH}}$, the GH hybrid (GHH) from [20]; 4) $\mathrm{edgeCost}_{\mathsf{SQSE}}$, the shape-quality-weighted SE (SQSE), which is newly proposed herein; and 5) $\mathrm{edgeCost}_{\mathsf{JNDSE}}$, the JND-weighted SE

(JNDSE), which is newly proposed herein. These preceding functions are defined as follows. The first function $\mathrm{edgeCost}_{\mathsf{D}}(T,e)$ is defined to be 0 if e passes the preferred-directions-augmented in-circle (Delaunay) test in [39], and 1 otherwise. The remaining functions are then given by

$$\mathrm{edgeCost}_{\mathsf{SE}}(T,e) = \beta(T,e),$$

$$\mathrm{edgeCost}_{\mathsf{GHH}}(T,e) = \begin{cases} [\mathrm{sq}(f_1)\,\mathrm{sq}(f_2)]^{-1} & \tau \leq \frac{1}{2} \\ \beta(T,e) & \text{otherwise,} \end{cases}$$

$$\mathrm{edgeCost}_{\mathsf{SQSE}}(T,e) = [\mathrm{sq}(f_1)\,\mathrm{sq}(f_2)]^{-1}\,\beta(T,e), \quad \text{and}$$

$$\mathrm{edgeCost}_{\mathsf{JNDSE}}(T,e) = \mathrm{edgeCost}_{\mathsf{JND}}(T,e)\beta(T,e), \quad \text{where}$$

$$\beta(T,e) = \sum_{p \in \boldsymbol{\mathcal{P}}_T(f_1) \cup \boldsymbol{\mathcal{P}}_T(f_2)} \left(\hat{\phi}_T(p) - \phi(p)\right)^2,$$

$\tau = \frac{\min\{\sigma,\sigma'\}}{\max\{\sigma,\sigma'\}}$, $\sigma = \mathrm{sq}(f_1)\,\mathrm{sq}(f_2)$, $\sigma' = \mathrm{sq}(f_1')\,\mathrm{sq}(f_2')$, and f_1' and f_2' are the two faces incident on the edge e', with e' denoting the edge obtained by applying an edge flip to e.

With the above edge-cost functions having been defined, we are now ready to introduce the twelve edge-flip criteria considered in our work, known by the names ABN, JND, DLP, DP, YMS, ELABN, ELJND, Delaunay, SE, GHH, SQSE, and JNDSE. Let e denote the edge to be tested for optimality in the triangulation T, where $e = \overline{v_i v_j}$. Let e' denote the new edge obtained by applying an edge-flip transformation to e, and let T' be the triangulation obtained if e is flipped. As mentioned above, we employ two different classes of edge-flip criterion in our work. We consider each in turn below.

The first class of edge-clip criterion assigns a cost to each edge in the triangulation, and then assigns a cost to the triangulation, which corresponds to the sum of the edge costs. The edge e is then preferred (over e') if the cost of triangulation T does not exceed the cost of triangulation T'. That is, the first class of edge-flip criterion chooses $\mathrm{isPreferred}$ to be of the form

$$\mathrm{isPreferred}(e) = \begin{cases} 1 & \mathrm{triCost}(T) \leq \mathrm{triCost}(T') \\ 0 & \text{otherwise,} \end{cases} \quad \text{where} \tag{5a}$$

$$\mathrm{triCost}(T) = \sum_{e \in \boldsymbol{\mathcal{E}}(T)} \mathrm{edgeCost}(T,e). \tag{5b}$$

The ABN, JND, DLP, DP, YMS, ELABN, and ELJND edge-flip criteria are obtained by choosing $\mathrm{isPreferred}$ as given in (5) with $\mathrm{edgeCost}$ selected as $\mathrm{edgeCost}_{\mathsf{ABN}}$, $\mathrm{edgeCost}_{\mathsf{JND}}$, $\mathrm{edgeCost}_{\mathsf{DLP}}$, $\mathrm{edgeCost}_{\mathsf{DP}}$, $\mathrm{edgeCost}_{\mathsf{YMS}}$, $\mathrm{edgeCost}_{\mathsf{ELABN}}$, and $\mathrm{edgeCost}_{\mathsf{ELJND}}$, respectively. Note that, although the summation in (5b) is taken over all edges in the triangulation, only a small number of terms differ between $\mathrm{triCost}(T)$

and $\mathrm{triCost}(T')$ for all of the edge-cost functions introduced above. So, as a practical matter, in order to compare $\mathrm{triCost}(T)$ and $\mathrm{triCost}(T')$, it is only necessary to compute the relatively small number of terms that can actually differ between the two summations.

The second class of edge-flip criterion simply uses the edge-cost function directly in order to decide whether to flip an edge. If the edge e has a strictly higher cost then its flipped counterpart e', the decision is made to flip the edge. That is, the second class of edge-flip criterion chooses $\mathrm{isPreferred}$ to be of the form

$$\mathrm{isPreferred}(e) = \begin{cases} 1 & \mathrm{edgeCost}(T, e) \leq \mathrm{edgeCost}(T', e') \\ 0 & \text{otherwise.} \end{cases} \tag{6}$$

The Delaunay, SE, GHH, SQSE, and JNDSE edge-flip criteria are obtained by choosing $\mathrm{isPreferred}$ as given in (6) with $\mathrm{edgeCost}$ selected as $\mathrm{edgeCost}_{\mathsf{D}}$, $\mathrm{edgeCost}_{\mathsf{SE}}$, $\mathrm{edgeCost}_{\mathsf{GHH}}$, $\mathrm{edgeCost}_{\mathsf{SQSE}}$, and $\mathrm{edgeCost}_{\mathsf{JNDSE}}$, respectively. Observe that, the edge-flip criteria JNDSE and SQSE, which are newly proposed herein, employ an underlying edge-cost function that weighs the squared error by some measure related to triangle shape (namely, shape quality or JND). Also, note that the use of the Delaunay edge-flip criterion in the LOP produces a Delaunay triangulation.

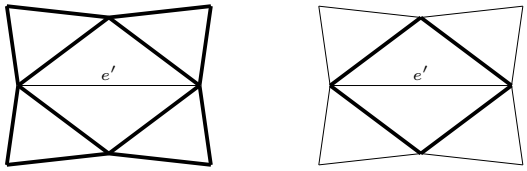
At this point, we would like to make one brief but important comment regarding the SE edge-flip criterion. From its definition above, observe that the SE edge-flip criterion will only choose to flip an edge if the squared approximation error, as defined in (1), is *strictly* reduced by the edge flip. Since the objective herein is to minimize this squared error, the reader might have the initial suspicion that consistently using the SE criterion everywhere in our framework must trivially lead to the best results (i.e., the lowest squared error). As we shall see later, however, this suspicion is, in fact, wrong.

Determination of suspect edges. Earlier, in the discussion of our mesh-generation framework and the LOP, the question arose as to which edges can become suspect when a new point is inserted in the triangulation or an edge is flipped. Although a general (and unavoidably vague) answer to this question was given at that time, we are now in a position to provide a much more precise answer to this question, which we do in what follows.

Recall that, in step 6 of our proposed mesh-generation framework (i.e., main connectivity adjustment), the LOP is performed after having inserted a new point p^* in the triangulation. When the LOP is invoked, we need to determine which edges should be initially marked as suspect. This, however, depends on the specific edge-flip criterion being employed. Let $\mathfrak{O}_0$ denote the set of all faces incident on the new vertex p^* and let $\mathfrak{O}_1$ denote the set of all faces that share at least one edge with a face in $\mathfrak{O}_0$. In the cases of the ABN, JND, DLP, DP, ELABN, ELJND, and YMS edge-flip criteria, the edges that should be marked as

(a) (b)

Fig. 4. Example of determining the newly suspect edges after the insertion of the point p^*. (a) First case. (b) Second case.



(a) (b)

Fig. 5. Example of determining the newly suspect edges after an edge flip that produces the edge e'. (a) First case. (b) Second case.

suspect are the flippable edges of all faces in $\mho_0 \cup \mho_1$. For example, after inserting the point p^* in a face, as shown in Fig. 4(a), the edges drawn using a thicker line would be marked as suspect. In the cases of the D, SE, GHH, SQSE, and JNDSE edge-flip criteria, the edges that should be marked as suspect are all flippable edges of all faces in $\mho_0$. For example, after inserting the point p^* in a face, as shown in Fig. 4(b), the edges drawn with a thicker line would be marked as suspect.

Also, recall that, in step 5 of the LOP, whenever an edge is flipped, we must determine which edges can become newly suspect as a result of the edge flip. Again, this depends on the particular edge-flip criterion being used. Let e denote the edge being flipped; let q denote the quadrilateral formed by the union of the two faces incident on e; and let e' denote the edge obtained by applying an edge flip to e. In the cases of the ABN, JND, DLP, DP, ELABN, ELJND, and YMS edge-flip criteria, the edges that should be marked as suspect are all flippable edges belonging to q or belonging to faces incident on edges of q. For example, if the edge e' as shown in Fig. 5(a) was just produced as a result of an edge flip, we would need to mark all of the edges drawn with a thicker line as suspect (presuming that each edge is flippable). In the case of the D, SE, GHH, SQSE, and JNDSE edge-flip criteria, the edges that should be marked as suspect are all flippable edges belonging to q. For example, if the edge e' as shown in Fig. 5(b) was just produced as a result of an edge flip, we would need to mark all of the edges drawn with a thicker line as suspect.

IV. Proposed Mesh-Generation Method and Its Development

So far, we have introduced our proposed mesh-generation framework, which has several free parameters, and suggested a number of possible choices for each of these parameters. As one might suspect, however, not all of these choices are equally good. In our work, we studied how different choices for the these parameters affect mesh quality. Because the best choice for one particular parameter can depend on how the remaining parameters are selected, our study needed to consider many parameter combinations. In

TABLE I

TEST IMAGES

Image	Size, Bits/Sample	Description
bull	1024×768, 8	computer-generated bull [42]
ct	512×512, 12	CT scan of head [40]
glasses2	1024×768, 8	raytraced glasses [42]
lena	512×512, 8	woman [41]
mri	256×256, 11	MRI scan of head [40]
peppers	512×512, 8	collection of peppers [41]

what follows, we present a summary of our analysis, by describing the general trends and providing some specific numerical results that well illustrate these trends. Based on our analysis, we recommend a particular best choice for the parameters, which leads to the specific mesh-generation method proposed herein.

Before proceeding further, a brief digression is in order regarding the image data that we used for analysis and evaluation purposes. In our work, we employed 42 images as test data, many of which were taken from standard image sets, such as [40] and [41]. Herein, we present results for a representative subset of these images, namely, those listed in Table I. This subset was deliberately chosen to contain a variety of image types including photographic, medical, and computer-generated imagery.

A. *Choice of Face- and Candidate-Selection Policies (*selFace *and* selCand*)*

First, we examine how the choice of the face-selection policy $\mathrm{selFace}$ affects mesh quality. To do this, we fix the candidate-selection policy $\mathrm{selCand}$ as PAE, the main-edge flip criterion $\mathrm{isPreferred}_{\mathsf{main}}$ as JNDSE, and the final connectivity adjustment as enabled with the final edge-flip criterion $\mathrm{isPreferred}_{\mathsf{final}}$ as SE. Then, we select from amongst the two face-selection policies under consideration, namely, GAE and GSE. For all 42 images in our test set and several sampling densities, we generated a mesh using each of the face-selection policies, and then measured the resulting approximation error in terms of PSNR. A representative subset of the results (namely, for the six images in Table I) is given in Table II, with the best result in each case (i.e., each row in the table) shown in boldface.

Since the goal in our work is to minimize the squared error as defined in (1), we would suspect that it is probably beneficial to choose the face with the greatest squared error into which to insert a new point (i.e., the GSE policy). Examining Table II, we can confirm this suspicion to be correct. As the results show, the GSE policy beats the GAE policy in all cases by a margin of 0.03 to 6.80 dB (with a median

of 1.945 dB). Clearly, the GSE policy is superior.

Although we have only shown results above for one specific choice of the fixed parameters (i.e., the parameters other than the face-selection policy), we generally found similar results with other choices. In passing, we also note that the computational complexity of the GSE and GAE policies are quite comparable. For the above reasons, we deem the GSE policy as best and recommend its use in our framework.

Next, we examine how the choice of the candidate-selection policy $\mathrm{selCand}$ affects mesh quality. To do this, we fix the face-selection policy $\mathrm{selFace}$ as GSE, the main edge-flip criterion $\mathrm{isPreferred}_{\mathsf{main}}$ as JNDSE, and the final connectivity adjustment as enabled with the final edge-flip criterion $\mathrm{isPreferred}_{\mathsf{final}}$ as SE. The candidate-selection policy is then chosen from amongst the three under consideration, namely, PAE, AMSE, and hybrid. For all 42 images in our test set and several sampling densities, we generated a mesh using each of the various candidate-selection policies and measured the resulting approximation error in terms of PSNR. A representative subset of the results (namely, for the six images in Table I) is given in Table III, with the best and worst values in each test case indicated by boldface and gray, respectively.

Examining Table III, we see that the AMSE and hybrid polices outperform the PAE policy in the vast majority of cases. In particular, the AMSE and hybrid policies beat the PAE policy in 20/24 and 23/24 of the test cases, respectively. We can also see that the hybrid policy outperforms the AMSE policy in 15/24 of the test cases. Furthermore, the hybrid policy typically wins by a much larger margin than it loses. That is, of the 9 cases where the AMSE policy beats the hybrid policy, only 2 are by more than a margin of 0.1 dB, whereas of the cases where the hybrid policy beats the AMSE policy, 13 involve a margin of more than 0.1 dB. From the above observations, it is clear that the hybrid policy is best.

Although we have only shown results above for one particular choice of the fixed parameters (i.e., the parameters other than the candidate-selection policy), we have generally found similar results with other choices. In passing, we note that, since the hybrid policy requires significantly less computation than the AMSE policy, selecting the former over the latter also happens to be beneficial in terms of computational complexity. For the reasons above, we deem the hybrid policy (which is newly proposed herein) to be most effective and recommend its use in our framework.

B. Choice of Main Edge-Flip Criterion $(\mathrm{isPreferred}_{\mathsf{main}})$

Next, we consider how mesh quality is affected by the choice of the main edge-flip criterion $\mathrm{isPreferred}_{\mathsf{main}}$ used in initial and main connectivity adjustment. To do this, we fix the face-selection policy $\mathrm{selFace}$ as

TABLE II

COMPARISON OF THE MESH QUALITY OBTAINED WITH THE VARIOUS FACE-SELECTION POLICIES

Image	Samp. Density (%)	PSNR (dB) GAE	GSE
bull	0.125	28.58	**35.38**
	0.250	35.70	**39.40**
	0.500	39.24	**41.42**
	1.000	41.85	**43.11**
ct	0.250	31.45	**33.66**
	0.500	36.85	**38.69**
	1.000	41.68	**43.09**
	2.000	46.36	**47.70**
glasses2	0.250	18.99	**22.75**
	0.500	22.70	**26.05**
	1.000	26.55	**29.99**
	2.000	31.40	**34.51**
lena	0.500	23.45	**26.51**
	1.000	27.37	**29.41**
	2.000	30.76	**32.03**
	3.000	32.84	**33.55**
mri	0.500	27.11	**31.23**
	1.000	33.44	**34.29**
	2.000	36.77	**37.56**
	3.000	39.18	**39.21**
peppers	0.250	21.70	**23.55**
	0.500	24.57	**26.83**
	1.000	28.47	**29.58**
	2.000	31.27	**31.70**

TABLE III

COMPARISON OF THE MESH QUALITY OBTAINED WITH THE VARIOUS CANDIDATE-SELECTION POLICIES

Image	Samp. Density (%)	PSNR (dB) PAE	AMSE	Hybrid
bull	0.125	35.38	36.41	**36.48**
	0.250	39.40	40.25	**40.45**
	0.500	41.42	42.63	**42.68**
	1.000	43.11	**44.32**	44.29
ct	0.250	33.66	**33.87**	33.85
	0.500	38.69	38.38	**38.70**
	1.000	43.09	42.70	**43.16**
	2.000	47.70	47.50	**47.82**
glasses2	0.250	22.75	**23.68**	23.56
	0.500	26.05	26.76	**26.87**
	1.000	29.99	30.08	**30.27**
	2.000	**34.51**	34.07	34.40
lena	0.500	26.51	**27.45**	27.37
	1.000	29.41	**30.22**	30.16
	2.000	32.03	32.66	**32.80**
	3.000	33.55	34.09	**34.21**
mri	0.500	31.23	31.89	**32.07**
	1.000	34.29	34.78	**35.17**
	2.000	37.56	37.69	**37.88**
	3.000	39.21	39.36	**39.59**
peppers	0.250	23.55	**24.71**	24.56
	0.500	26.83	**27.81**	27.73
	1.000	29.58	**30.56**	30.52
	2.000	31.70	**32.64**	32.63

GSE, the candidate-selection policy $\mathrm{selCand}$ as hybrid, and the final connectivity adjustment as disabled. Then we select $\mathrm{isPreferred}_{\mathsf{main}}$ from amongst the various criteria under consideration, namely, D, ABN, JND, DLP, DP, ELABN, ELJND, YMS, SE, GHH, SQSE, and JNDSE. For all 42 images in our test set and several sampling densities, we generated a mesh using each of the various main edge-flip criteria under consideration, and then measured the resulting approximation error in terms of PSNR. A representative subset of the results obtained (namely, for the six images in Table I) is given in Table IV, where the best

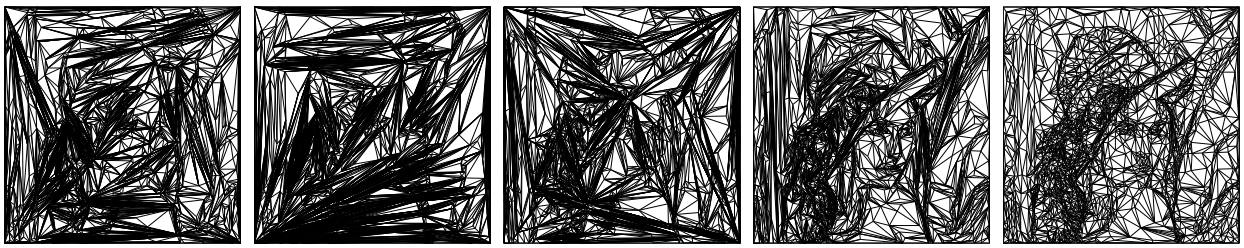
TABLE IV

COMPARISON OF THE MESH QUALITY OBTAINED WITH THE VARIOUS MAIN EDGE-FLIP CRITERIA

Image	Samp. Density (%)	PSNR (dB)											
		D	ABN	JND	DLP	DP	ELABN	ELJND	YMS	SE	GHH	SQSE	JNDSE
bull	0.125	31.60	27.50	34.17	33.16	23.55	31.70	33.00	29.30	26.86	34.51	35.62	**35.75**
	0.250	35.92	32.66	38.63	38.43	27.17	37.20	37.96	34.97	32.60	39.08	39.76	**39.84**
	0.500	39.59	37.04	41.21	41.42	31.79	40.41	40.78	39.56	37.29	41.90	42.17	**42.18**
	1.000	41.98	40.74	42.80	42.92	35.90	42.44	42.61	42.01	41.22	43.67	**43.89**	43.88
ct	0.250	30.51	24.66	31.30	30.02	22.96	28.07	31.46	27.74	27.03	32.39	33.29	**33.40**
	0.500	35.40	26.97	36.64	35.53	25.18	33.23	36.66	33.06	30.80	37.56	38.10	**38.24**
	1.000	39.47	30.63	41.48	40.25	28.05	38.17	41.27	38.87	34.90	42.06	42.57	**42.74**
	2.000	43.78	34.90	46.34	45.36	31.14	43.63	46.01	44.08	38.95	46.78	47.39	**47.42**
glasses2	0.250	21.60	18.34	21.37	20.74	16.65	20.07	21.52	19.68	18.45	22.93	**23.17**	23.14
	0.500	24.42	20.25	24.52	23.68	18.42	22.80	24.43	22.02	20.51	25.87	26.21	**26.43**
	1.000	27.43	22.53	28.09	27.23	20.17	25.83	27.87	25.15	22.89	29.17	29.61	**29.81**
	2.000	30.97	25.30	32.32	31.40	22.23	29.70	31.83	29.34	25.43	33.18	33.78	**33.95**
lena	0.500	25.08	21.06	25.40	24.89	19.69	23.57	25.16	23.84	23.05	26.39	**26.92**	26.89
	1.000	27.82	23.44	28.35	27.90	21.59	26.22	27.88	26.53	25.63	29.20	29.66	**29.70**
	2.000	30.37	26.00	31.05	30.54	23.89	29.07	30.79	29.45	28.11	31.96	32.29	**32.39**
	3.000	31.86	27.70	32.56	32.14	25.38	30.84	32.31	31.10	29.77	33.45	33.74	**33.83**
mri	0.500	29.91	25.37	29.69	29.18	25.04	27.56	29.89	28.11	27.65	31.20	31.37	**31.62**
	1.000	32.75	27.84	33.13	32.51	27.14	30.89	33.10	31.06	30.01	34.18	34.52	**34.79**
	2.000	35.80	30.74	36.06	35.39	29.24	34.12	36.10	34.12	32.70	37.18	37.48	**37.54**
	3.000	37.47	32.31	37.84	37.03	30.30	36.16	37.97	36.07	34.70	38.81	39.21	**39.24**
peppers	0.250	22.36	18.33	22.18	21.53	16.72	19.98	22.53	19.52	20.31	23.85	24.19	**24.19**
	0.500	25.53	21.15	25.73	24.82	18.63	24.14	25.73	23.17	23.06	27.02	27.27	**27.30**
	1.000	28.31	24.07	28.62	28.22	20.83	27.11	28.69	26.68	26.30	29.88	29.99	**30.14**
	2.000	30.85	27.02	31.10	30.76	23.25	29.84	31.13	29.68	28.76	32.08	32.18	**32.28**

and worst values in each test case are again shown in boldface and gray, respectively.

Let us now examine Table IV. Rather than attempting to individually rank each of the criteria from best to worst (which may not be possible to do in general), we instead observe that there are some clear winners and losers here. In particular, the best performers are the JNDSE, SQSE, and GHH criteria. Of these three, the JNDSE criterion (which is newly proposed herein) is the overall winner, yielding the best result in 21/24 of the test cases and never losing by a margin of more than 0.03 dB in the remaining cases. At the other extreme, the worst performers in decreasing order of badness are the DP, and ABN



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

Fig. 6. The triangulations obtained for the `lena` image at a sampling density of 1% with the main edge-flip criterion chosen as (a) ABN, (b) DP, (c) SE, (d) DLP, and (e) JNDSE, respectively.

and SE criteria, which produce the three poorest results in every case. The DP criterion is the worst performer overall, yielding the poorest result in all of the test cases. The ABN criterion is second worst in 22/24 of the test cases. The SE criterion is second worst or third worst in 22/24 of the test cases. The poor performance of the SE criterion is extremely important to note. It is because the SE edge-flip criterion performs so badly that the intuitively "obvious" solution of simply always using the SE edge-flip criterion in our framework is not a good one.

Although we have only shown results above for one particular choice of the fixed parameters (i.e., the parameters other than the main edge-flip criterion), we have generally found similar results with other choices. Since the JNDSE criterion is the clear winner, we recommend its use as the main edge-flip criterion $\mathrm{isPreferred}_{\mathsf{main}}$ in our framework.

A more careful examination of the results shows that the DP, SE, and ABN criteria perform very poorly due to their propensity to lead to triangulations with a very large number of *poorly chosen* sliver triangles. For example, for one of the test cases from Table IV, Fig. 6 shows the triangulations that were obtained in the cases of the ABN, DP, and SE criteria, which perform poorly, as well as the DLP and JNDSE criteria, which perform relatively better. Notice how the results for the ABN, DP, and SE criteria are dominated by many poorly chosen sliver triangles, whereas the DLP and JNDSE results are not. The good performance of the JNDSE, SQSE, and GHH edge-flip criteria can be explained by the fact that these three criteria take into account not only squared error but triangle shape as well (via shape quality in the GHH and SQSE cases and JND in the JNDSE case). This allows these criteria to avoid triangulations with many poorly chosen sliver triangles.

Although the interplay between edge-flip criteria and sliver triangles is often extremely complex, we are able to offer some explanation as to why the DP and SE edge-flip criteria (i.e., two of the three worst performers) tend to produce many sliver triangles. First, let us consider the SE criterion. Since a

sliver triangle has a very small area and therefore contains few or no points from the image domain Λ, the squared error calculated over a sliver triangle will always be very small and often zero. So, when viewed strictly from the standpoint of squared error, sliver triangles seem quite good, as they have very low squared error. Unfortunately, because of this fact, many undesirable situations can arise when the SE criterion is employed. For example, consider an edge e whose two incident faces f_1 and f_2 are each sliver triangles that contain no points from Λ. The SE criterion will never flip such an edge (i.e., $\mathrm{isPreferred}(e)$ will always be one). Such behavior can, in turn, increase the likelihood that the two sliver triangles f_1 and f_2 remain in the triangulation. This problem is further exacerbated by the fact that since f_1 and f_2 do not contain any points from Λ, no new point can be inserted into f_1 or f_2.

Next, let us contemplate why the DP criterion can lead to many sliver triangles. Consider an edge e with two incident faces f_1 and f_2 whose union forms the (strictly convex) quadrilateral q, and let e' denote the edge obtained by flipping e. Suppose that e and e' differ greatly in length, say, by a factor of more than 64, so that q is long and thin and f_1 and f_2 are sliver triangles. Let e_{short} and e_{long} denote the shorter and longer of the edges e and e', respectively. Due to the manner in which the DP edge-cost function $\mathrm{edgeCost}_{\mathsf{DP}}$ is defined in (4d), it is statistically very likely that $\mathrm{edgeCost}_{\mathsf{DP}}(e_{\mathsf{long}})$ is quite small and $\mathrm{edgeCost}_{\mathsf{DP}}(e_{\mathsf{long}}) < \mathrm{edgeCost}_{\mathsf{DP}}(e_{\mathsf{short}})$. Thus, the DP edge-cost function tends to strongly favor longer edges that are associated with sliver triangles. Because the DP edge-flip criterion is based on this edge-cost function, all others things being equal, the DP edge-flip criterion will tend to prefer longer edges when they are associated with sliver triangles, and this in turn tends to lead to triangulations with many sliver triangles.

C. Choice of Final Edge-Flip Criterion ($\mathrm{isPreferred}_{\mathsf{final}}$)

Next, we examine how the choice of the final edge-flip criterion $\mathrm{isPreferred}_{\mathsf{final}}$ affects mesh quality. To do this, we fix the face-selection policy $\mathrm{selFace}$ as GSE, the candidate-selection policy $\mathrm{selCand}$ as hybrid, and the main edge-flip criterion $\mathrm{isPreferred}_{\mathsf{main}}$ as Delaunay. Then, we select from amongst the cases of no final connectivity adjustment as well as final connectivity adjustment with the final edge-flip criterion $\mathrm{isPreferred}_{\mathsf{final}}$ chosen as each of Delaunay, ABN, JND, DLP, DP, ELABN, ELJND, YMS, SE, GHH, SQSE, and JNDSE. For all 42 images in our test set and several sampling densities, we generated a mesh using each of the schemes under consideration, and then measured the resulting approximation error in terms of PSNR. A representative subset of the results obtained is given in Table V, with the best result in each test case shown in boldface. The case of no final connectivity adjustment is labelled as "None" in the table. Note that, we exclude the case of choosing $\mathrm{isPreferred}_{\mathsf{final}}$ as Delaunay since this is equivalent to no final connectivity adjustment when $\mathrm{isPreferred}_{\mathsf{main}}$ is Delaunay.

TABLE V

COMPARISON OF THE MESH QUALITY OBTAINED WITH THE VARIOUS FINAL EDGE-FLIP CRITERIA

Image	Samp. Density (%)	PSNR (dB)											
		None	ABN	JND	DLP	DP	ELABN	ELJND	YMS	SE	GHH	SQSE	JNDSE
bull	0.125	31.60	30.62	30.01	31.06	30.22	30.91	31.40	28.81	**34.14**	32.65	32.77	32.88
	0.250	35.92	34.55	35.05	36.08	34.57	35.44	36.00	34.22	**38.37**	35.81	37.09	37.16
	0.500	39.59	38.68	38.98	39.30	38.62	39.20	39.63	37.47	**41.53**	39.98	40.45	40.48
	1.000	41.98	41.08	41.29	41.63	41.09	41.46	41.85	40.14	**43.58**	42.68	42.78	42.85
ct	0.250	30.51	27.63	28.86	29.09	26.17	28.26	30.17	26.63	**32.16**	31.23	31.63	31.64
	0.500	35.40	32.87	32.86	34.43	31.26	33.69	35.46	33.58	**36.97**	32.54	36.27	36.37
	1.000	39.47	36.66	38.41	38.40	35.47	36.98	39.50	36.96	**41.18**	39.54	40.45	40.51
	2.000	43.78	41.89	41.10	42.25	40.93	42.71	44.05	41.19	**45.86**	40.85	45.01	45.13
lena	0.500	25.08	23.30	24.00	24.39	23.49	23.86	24.75	23.54	**26.50**	25.91	25.84	25.92
	1.000	27.82	26.00	27.19	27.13	25.74	26.37	27.55	26.32	**29.24**	28.71	28.63	28.73
	2.000	30.37	28.83	29.92	29.80	28.70	29.17	30.21	29.08	**31.81**	31.26	31.18	31.27
	3.000	31.86	30.28	31.39	31.30	30.14	30.66	31.70	30.63	**33.31**	32.67	32.63	32.72

Examining Table V, we can see that the SE criterion performs best in every test case, followed by the JNDSE and SQSE criteria (in that approximate order). Moreover, the use of final connectivity adjustment with the SE criterion beats the case of no final connectivity adjustment (i.e., the "None" column) by a margin of 1.42 to 2.54 dB. So, clearly the use of final connectivity adjustment with the SE criterion is extremely beneficial. Although we have only presented results for one set of choices for the fixed parameters (i.e., the parameters other than the final edge-flip criterion), the preceding trends were found to be followed for other choices, provided that the main edge-flip criterion was chosen to produce a reasonably good triangulation as input to final connectivity adjustment. In cases where the main edge-flip criterion produces very poor triangulations (i.e., the DP, SE, and ABN cases), it actually turns out that the JNDSE and SQSE criteria perform better than the SE criterion.

The above behavior can be explained as follows. Nominally, we would expect the SE criterion to perform best, as it directly minimizes the squared error in (1). In cases where the triangulation that is input to the final connectivity adjustment process is reasonably good, this is the exactly the behavior observed. If, however, the main edge-flip criterion is chosen as one of the criteria (i.e., DP, SE, and ABN) that leads to very poor triangulations, the behavior changes with the JNDSE and SQSE criteria becoming much more effective than the SE criterion. This change in behavior is due to the fact that the SE criterion does not take triangle shape into account. If the input triangulation (for final connectivity adjustment)

has many badly shaped triangles, the SE criterion cannot improve much upon the situation since it is effectively blind to triangle shape, which will ultimately lead to a poor quality mesh. In contrast, the JNDSE and SQSE criteria both consider triangle shape (in addition to squared error). For this reason, they are able to partially mitigate the effects of a poorly chosen triangulation and produce a better result than is possible with the SE criterion.

Since the main edge-flip criterion $\mathrm{isPreferred}_{\mathsf{main}}$ was previously recommended to be chosen as JNDSE, which generally produces good triangulations with very few poorly-chosen sliver triangles, we conclude that we should use final connectivity adjustment with a final edge-flip criterion that works well for good triangulations. As we saw above, the clear choice in this case is the SE criterion. Therefore, in our framework, we recommend always using final connectivity adjustment and selecting the final edge-flip criterion $\mathrm{isPreferred}_{\mathsf{final}}$ as the SE criterion.

D. Proposed Mesh-Generation Method

Above, we have studied how various choices of the free parameters of our proposed framework affect mesh quality. This led us to recommend a particular choice for each of these parameters. The mesh-generation method that we propose in this paper simply corresponds to our framework with the parameters selected as recommended above, namely, with the face-selection policy $\mathrm{selFace}$ as GSE, the candidate-selection policy $\mathrm{selCand}$ as hybrid, the main edge-flip criterion $\mathrm{isPreferred}_{\mathsf{main}}$ as JNDSE, and final connectivity adjustment enabled with the final edge-flip criterion $\mathrm{isPreferred}_{\mathsf{final}}$ as SE.

V. Evaluation of Proposed Mesh-Generation Method

Having introduced our proposed method, we now evaluate its performance by comparing it to several other competing schemes in terms of mesh quality and computational/memory complexity. The first of the methods that we consider for comparison purposes is the one proposed by Garland and Heckbert [20, Algorithm IV] with the quality threshold parameter `qthresh` chosen as 0.5 and an L^2 error measure, which we henceforth refer to by the name "GH". The GH method is essentially a special case of our framework with no final connectivity adjustment and the face-selection policy, candidate-selection policy, and main edge-flip criterion chosen as GAE, PAE, and GHH, respectively. The second of the methods that we consider for comparison purposes is the one proposed by Rippa in [22] with the least-squares edge cost (in the interpolating case), which we henceforth refer to by the name "R". (As an aside, we note that we elected to consider the least-squares edge-cost function from [22], as it is the only edge-cost function therein that incorporates squared error in some way.) The R method is a special case of our framework with no final connectivity adjustment and the face-selection policy, candidate-selection policy,

and main edge-flip criterion chosen as GAE, PAE, and SE, respectively. The third of the methods that we consider for comparison purposes is the adaptive-thinning (AT) scheme of Demaret and Iske [43]. The AT method is one of the very best state-of-the-art Delaunay-based methods. It produces extremely high-quality meshes, even better than many DDT-based schemes, but has extremely high computational and memory requirements.

Mesh quality. For all 42 images in our test set and several sampling densities, we used each of the methods under consideration to generate a mesh and then measured the resulting approximation error in terms of PSNR. A representative subset of the results obtained (namely, for the six images in Table I) is shown in Table VI, with the best result in each case (i.e., each row in the table) highlighted in boldface. In order to show that the improvements offered by our proposed method are not solely due to its use of the GSE face-selection scheme, we also consider our own improved versions of the GH and R methods, called GH2 and R2, respectively, where the face-selection policy of GAE is replaced by GSE.

Examining Table VI, we can see that our proposed method is the clear winner, producing the best result in all 24 test cases. Inspecting the results more closely, we find that our proposed method outperforms the GH, R, GH2, R2, and AT schemes by margins of 1.58 to 9.93 dB (with median 4.1050 dB), 7.86 to 17.37 dB (with median 10.765 dB), 1.02 to 3.36 dB (with median 1.5800 dB), 4.16 to 10.65 dB (with median 6.3650 dB), and 0.20 to 3.36 dB (with median 0.8250 dB), respectively. The fact that our proposed method beats the GH2 and R2 schemes (i.e., our improved versions of the GH and R schemes, respectively) by significant margins demonstrates that the excellent results from our method are not simply due to our different point-selection strategy alone. The fact that our method can outperform the AT scheme is extremely impressive, given that the AT scheme produces very high quality meshes and (as we shall see shortly) requires over 10 times more computation and orders of magnitude more memory. Again, examining Table VI, we observe that the R and R2 methods are the clear losers, yielding the two worst results in every case.

In the above evaluation, PSNR was found to correlate reasonably well with subjective quality. For the benefit of the reader, however, we provide an example illustrating the subjective quality achieved by the various methods. In particular, for one of the test cases involving the `bull` image (from Table VI), a small part of each image reconstruction is shown under magnification in Fig. 7, along with the corresponding triangulation. Examining the figure, we can see that our proposed method produces approximations with better subjective quality as compared to the other methods. So, our proposed method not only yields approximations with low squared error, but good subjective quality as well. Observe that the R and R2 methods perform very poorly, due to the large number of poorly chosen sliver triangles in their

TABLE VI

COMPARISON OF THE MESH QUALITY OBTAINED WITH THE VARIOUS MESH-GENERATION METHODS

Image	Samp. Density (%)	PSNR (dB)					
		Proposed	GH	R	GH2	R2	AT
bull	0.125	**36.48**	26.55	19.26	33.12	25.83	33.12
	0.250	**40.45**	31.99	23.08	38.28	30.59	38.23
	0.500	**42.68**	37.64	26.46	40.72	35.81	41.87
	1.000	**44.29**	40.56	32.40	42.48	40.13	43.99
ct	0.250	**33.85**	28.69	20.75	32.22	26.02	32.15
	0.500	**38.70**	35.23	25.92	37.68	29.61	37.22
	1.000	**43.16**	39.43	30.08	42.01	33.91	41.35
	2.000	**47.82**	44.99	36.16	46.61	39.12	45.33
glasses2	0.250	**23.56**	18.38	14.07	21.94	17.51	23.36
	0.500	**26.87**	20.95	15.29	25.07	19.19	25.84
	1.000	**30.27**	25.27	17.82	28.87	21.38	28.88
	2.000	**34.40**	29.94	20.99	33.11	24.26	32.73
lena	0.500	**27.37**	22.39	16.66	25.37	21.93	26.66
	1.000	**30.16**	25.27	19.34	28.51	24.20	29.12
	2.000	**32.80**	29.80	22.41	31.26	27.03	31.82
	3.000	**34.21**	31.84	24.11	32.77	28.77	33.37
mri	0.500	**32.07**	26.00	22.29	30.57	25.51	31.48
	1.000	**35.17**	32.38	24.48	33.77	28.34	34.39
	2.000	**37.88**	35.35	28.87	36.80	31.71	37.25
	3.000	**39.59**	38.01	31.73	38.42	33.80	39.01
peppers	0.250	**24.56**	19.84	14.78	22.58	19.20	23.93
	0.500	**27.73**	23.98	17.36	25.95	21.66	27.09
	1.000	**30.52**	27.49	20.32	28.77	24.60	30.05
	2.000	**32.63**	30.36	24.53	31.14	27.73	32.39

triangulations.

Computational complexity. Next, we compare the various mesh-generation methods in terms of their computational complexity (i.e., execution time). To do this, we provide a representative subset of some timing results collected on very modest hardware (namely, a seven year old notebook computer with a 3.4 GHz Intel Pentium 4 and 1 GB of RAM). For the lena image and several sampling densities, the time required for mesh generation for each of the methods under consideration is shown in Table VII. In the case of the GH and R methods, a second set of numbers is given in parentheses. These numbers

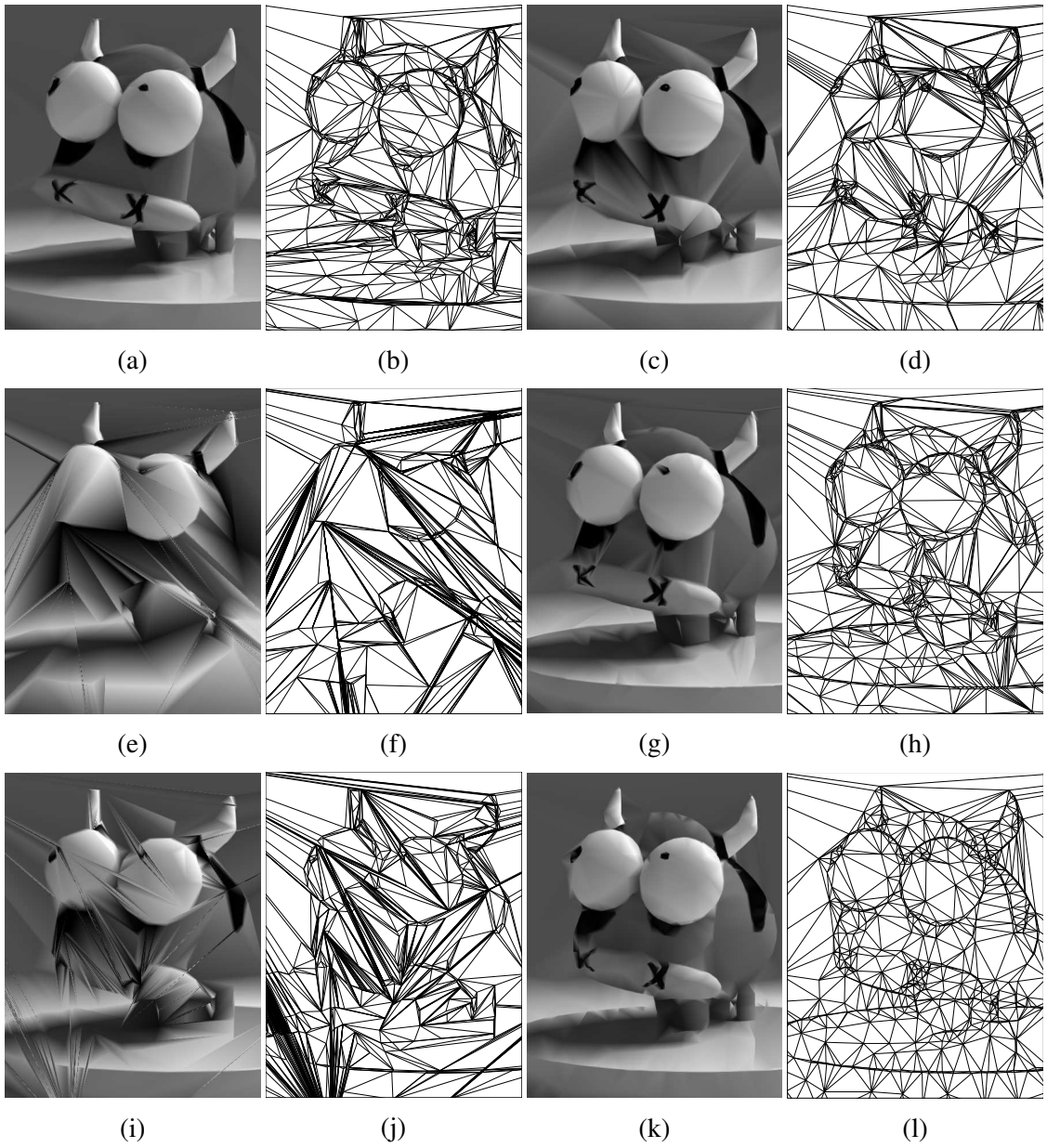


Fig. 7. Part of the image approximation obtained for the `bull` image at a sampling density of 0.125% with the (a) proposed (36.48 dB), (c) GH (26.55 dB), (e) R (19.26 dB), (g) GH2 (33.12 dB), (i) R2 (25.83 dB), and (k) AT (33.12 dB), methods and (b), (d), (f), (h), (j), and (l) their corresponding triangulations.

correspond to the time required if the method in question terminates mesh generation not when the target sampling density is achieved, but rather when the PSNR mesh quality matches that of the corresponding result from our proposed method.

Examining Table VII (excluding the results in parentheses), we observe the following. Our proposed method has very substantially lower complexity than the AT scheme, requiring only about 5 to 10% of

TABLE VII

COMPARISON OF THE COMPUTATIONAL COMPLEXITY FOR THE VARIOUS METHODS

Samp. Density (%)	Time (s)			
	Proposed	GH	R	AT
0.5	2.31	2.10 (2.85)	1.52 (4.22)	43.03
1.0	2.77	2.59 (3.33)	1.93 (5.45)	43.03
2.0	3.54	3.26 (4.10)	2.54 (7.25)	42.40
3.0	4.19	3.76 (4.67)	3.06 (8.53)	42.12

the time of the AT scheme. This is particularly impressive when one considers that our method produces significantly higher quality meshes than the AT scheme. Also, our proposed method requires about 6 to 12% and 36 to 52% more computation than the GH and R schemes, respectively. The increase in time relative to the GH and R schemes is due primarily to the additional final connectivity adjustment step and the use of a more computationally-expensive point-selection strategy (which often requires the computation of $\mathrm{selCand}_{\mathsf{AMSE}}$). When interpreting these computational-cost results, it is important to keep in mind that the GH and R schemes produce meshes of very significantly lower quality than the proposed method. If in the case of the GH and R methods, we instead let the mesh-generation process terminate when it achieves the same PSNR mesh quality as the proposed method, the execution times in parenthesis in Table VII apply. Viewing the situation from this perspective, the GH and R methods are actually much slower, by factors of 1.11 to 1.24 and 1.82 to 2.05, respectively. With the preceding observation in mind (as well as others made above), we deem the computational complexity of our scheme to be at least comparable to (and arguably better than) the GH and R methods. So overall, our proposed method performs quite well in terms of computational complexity.

The curious reader might wonder why the execution times of the GH and R schemes are not closer. As it turns out, the main reason for this difference is that the R scheme requires many fewer edge flips than the GH scheme, typically about 2.4 to 2.9 times less. Recall that the R scheme chooses a relatively large number of poorly chosen sliver triangles. This tends to significantly reduce the number of flippable edges, causing the LOP to converge much more quickly with fewer edge flips. So, essentially, the only reason that the R scheme is faster is because of the poor choice of triangulation that it yields.

Memory complexity. For all of the methods under consideration, memory usage is dominated by the triangulation data structure and a few auxiliary data structures (such as priority queues) whose size grows (approximately) proportionally to mesh size. Therefore, memory usage is essentially determined by mesh

size, and correspondingly, the peak memory usage is determined by the peak mesh size. For an image of width W, height H, and a given sampling density D, the peak mesh size for each of the proposed, GH, and R methods is DWH, whereas the peak mesh size for the AT scheme is WH (since the AT scheme begins with a mesh containing all WH sample points from the original image). So, the peak memory usage for the proposed, GH, and R methods is essentially the same, while, for values of D of practical interest, say $D \in [0.125\%, 3\%]$, the AT method requires 33 to 800 times more memory than the proposed, GH, and R methods. So, in terms of memory usage, our proposed method compares very favorably, being tied for best (i.e., lowest peak memory usage) with the GH and R schemes and requiring orders of magnitude less memory than the AT scheme.

VI. Conclusions

In this paper, we have proposed a DDT-based mesh-generation framework for image representation, derived by making a number of key modifications to the Rippa and Garland-Heckbert frameworks, including: 1) the addition of a final connectivity-adjustment step, 2) the development of more effective edge-flip criteria for the LOP, and 3) the introduction of a better point-selection strategy. Furthermore, we have also proposed a specific mesh-generation method, obtained by carefully choosing the free parameters of our framework. Through experimental results, our proposed mesh-generation method was shown to produce meshes of significantly higher quality (both in terms of squared error and subjectively) than those generated by the three competing schemes, namely, the GH, R, and AT schemes. In terms of computational cost, our proposed method was found to be at least comparable to the GH and R schemes (for the reasons given earlier) and to require over an order of magnitude less computation time than the AT method. In terms of memory cost, our proposed method was shown to require essentially the same amount of memory as the GH and R schemes and orders of magnitude less memory than the AT scheme. In short, our proposed method yields meshes of very high quality at reasonably low computational and memory costs. Consequently, our method is of great benefit to the many applications that employ mesh models of images. Furthermore, by further exploring the algorithmic possibilities afforded by our proposed framework, we are optimistic that even more effective mesh-generation schemes can be synthesized.

References

[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*, vol. 1, 2002, pp. 844–847.

[2] M. Petrou, R. Piroddi, and A. Talebpour, "Texture recognition from sparsely and irregularly sampled data," *Computer Vision and Image Understanding*, vol. 102, pp. 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*, Pittsburgh, PA, USA, Nov. 2007, pp. 235–241.

[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*, vol. 2, 2003, pp. 997–1000.

[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*, vol. 23, no. 2, pp. 202–212, Feb. 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*, vol. 1, Vancouver, BC, Canada, 2000, pp. 97–100.

[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*, 2003, pp. 16–23.

[8] ——, "Image interpolation by pixel-level data-dependent triangulation," *Computer Graphics Forum*, vol. 23, no. 2, pp. 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*, Taipei, Taiwan, Apr. 2009, pp. 1017–1020.

[10] G. Ramponi and S. Carrato, "An adaptive irregular sampling algorithm and its application to image coding," *Image and Vision Computing*, vol. 19, pp. 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*, vol. 2, 1997, pp. 724–727.

[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*, vol. 6, no. 6, pp. 647–659, Dec. 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*, vol. 5, no. 2, pp. 338–346, Feb. 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*, vol. 150, no. 1, pp. 44–50, Feb. 2003.

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

[16] I. Amidror, "Scattered data interpolation methods for electronic imaging systems: a survey," *Journal of Electronic Imaging*, vol. 11, no. 2, pp. 157–176, Apr. 2002.

[17] R. Franke and G. M. Nielson, "Scattered data interpolation and applications: A tutorial and survey," in *Geometric Modeling: Methods and Applications*. Springer-Verlag, 1991, pp. 131–160.

[18] B. Delaunay, "Sur la sphere vide," *Bulletin of the Academy of Sciences of the USSR, Classe des Sciences Mathematiques et Naturelle*, vol. 7, no. 6, pp. 793–800, 1934.

[19] S. Rippa, "Long and thin triangles can be good for linear interpolation," *SIAM Journal on Numerical Analysis*, vol. 29, no. 1, pp. 257–270, 1992.

[20] M. Garland and P. S. Heckbert, "Fast polygonal approximation of terrains and height fields," School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA, Tech. Rep. CMU-CS-95-181, Sep. 1995.

[21] ——, "Fast triangular approximation of terrains and height fields," draft manuscript, 1997, dated May 2, 1997 (19 pages).

[22] S. Rippa, "Adaptive approximation by piecewise linear polynomials on triangulations of subsets of scattered data," *SIAM Journal on Scientific and Statistical Computing*, vol. 13, no. 5, pp. 1123–1141, 1992.

[23] N. Dyn, D. Levin, and S. Rippa, "Data dependent triangulations for piecewise linear interpolation," *IMA Journal of Numerical Analysis*, vol. 10, pp. 137–154, 1990.

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

[25] E. Quak and L. L. Schumaker, "Least squares fitting by linear splines on data dependent triangulations," in *Curves and Surfaces*, P. J. Laurent, A. L. Mehaute, and L. L. Schumaker, Eds. Boston, MA, USA: Academic Press, 1991, pp. 387–390.

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

[27] J. Weisz and R. Bodnar, "A refined "angle between normals" criterion for scattered data interpolation," *Computers and Mathematics with Applications*, vol. 41, pp. 531–534, 2001.

[28] X. Yu, B. S. Morse, and T. W. Sederberg, "Image reconstruction using data-dependent triangulation," *IEEE Computer Graphics and Applications*, vol. 21, no. 3, pp. 62–68, May 2001.

[29] N. Dyn, D. Levin, and S. Rippa, "Algorithms for the construction of data dependent triangulations," in *Algorithms for Approximation II*, J. C. Mason and M. G. Cox, Eds. London: Chapman and Hall, 1990, pp. 185–192.

[30] C. L. Lawson, "Software for C^1 surface interpolation," in *Mathematical Software III*, J. R. Rice, Ed. New York, NY, USA: Academic Press, 1977, pp. 161–194.

[31] J. L. Brown, "Vertex based data dependent triangulations," *Computer Aided Geometric Design*, vol. 8, pp. 239–251, 1991.

[32] L. L. Schumaker, "Computing optimal triangulations using simulated annealing," *Computer Aided Geometric Design*, vol. 10, pp. 329–345, 1993.

[33] O. Kreylos and B. Hamann, "On simulated annealing and the construction of linear spline approximations for scattered data," *IEEE Trans. on Visualization and Computer Graphics*, vol. 7, no. 1, pp. 17–31, Jan. 2001.

[34] B. Lehner, G. Umlauf, and B. Hamann, "Image compression using data-dependent triangulations," *Lecture Notes in Computer Science*, vol. 4841, pp. 351–362, 2007.

[35] P. K. Agarwal and S. Suri, "Surface approximation and geometric partitions," in *Proc. of ACM-SIAM Symposium on Discrete Algorithms*, Jan. 1994, pp. 24–33.

[36] C. L. Lawson, "Transforming triangulations," *Discrete Mathematics*, vol. 3, pp. 365–372, 1972.

[37] E. Osherovich and A. M. Bruckstein, "All triangulations are reachable via sequences of edge-flips: an elementary proof," *Computer Aided Geometric Design*, vol. 25, pp. 157–161, 2008.

[38] K. Fleischer and D. Salesin, "Accurate polygon scan conversion using half-open intervals," in *Graphics Gems III*, 1995, pp. 362–365.

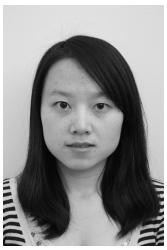
[39] C. Dyken and M. S. Floater, "Preferred directions for resolving the non-uniqueness of Delaunay triangulations," *Computational Geometry—Theory and Applications*, vol. 34, pp. 96–101, 2006.

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

[41] "USC-SIPI image database," 2011. [Online]. Available: http://sipi.usc.edu/database

[42] "Michael Adams' research datasets," 2011. [Online]. Available: http://www.ece.uvic.ca/~mdadams/datasets

[43] L. Demaret and A. Iske, "Advances in digital image compression by adaptive thinning," in *Annals of the Marie-Curie Fellowship Association*. Marie Curie Fellowship Association, Feb. 2004, vol. 3, pp. 105–109.



Ping Li received her B.A.Sc degree in biomedical engineering from Xi'an Jiaotong University, Xi'an, Shaanxi, China, and her M.A.Sc degree in electrical engineering from the University of Victoria, Victoria, BC, Canada. She is currently with Qualcomm Canada Inc., Markham, ON, Canada. Her research interests include signal/image processing, computational geometry, and algorithms.

Michael D. Adams (S'88–M'03–SM'09) received the B.A.Sc. degree in computer engineering from the University of Waterloo, Waterloo, ON, Canada, the M.A.Sc. degree in electrical engineering from the University of Victoria, Victoria, BC, Canada, and the Ph.D. degree in electrical engineering from the University of British Columbia, Vancouver, BC, Canada. Since 2003, Michael has been with the Department of Electrical and Computer of Engineering at the University of Victoria, Victoria, BC, Canada, where he is currently an Associate Professor. His research interests include signal processing, wavelets, data compression, multimedia systems, computational geometry, and algorithms.