Image Representation Using Triangle Meshes with Explicit Discontinuities*

Xi Tu and Michael D. Adams
Dept. of Electrical and Computer Engineering
University of Victoria
Victoria, BC, V8W 3P6, Canada
tuxi@ece.uvic.ca and mdadams@ece.uvic.ca

Abstract

A new mesh model for images, which explicitly represents discontinuities (i.e., image edges), is proposed along with a corresponding mesh-generation method that determines the mesh-model parameters for a given input image. The proposed approach is demonstrated to be capable of producing mesh-based image approximations of higher quality (both objectively and subjectively) than those generated by the state-of-the-art error-diffusion (ED) method of Yang, Wernick, and Brankov. For example, the image approximations produced by the proposed method are often about 2.5 dB higher in PSNR than those produced by the ED scheme.

1. Introduction

In recent years, there has been a growing interest in image representations that are based on triangle meshes. Such representations are beneficial in a number of applications, including image coding, motion tracking/compensation [5], and geometric image manipulation [4]. Different mesh models are characterized by different sets of model parameters. In order to use a mesh model in a particular application, we typically need a way in which to choose the model parameters for a given input image. That is, a mesh-generation method is needed. Many mesh-generation methods (along with their corresponding mesh models) have been proposed to date, with one highly effective scheme being the error-diffusion (ED) scheme of Yang et al. [5]. In this paper, we first introduce a mesh model that explicitly represents image edges. Then, we propose a mesh-generation method, called the **TA method**, that selects the model parameters for a given input image. As will be seen later, our approach produces image approximations of much better quality than those produced by the highly effective ED scheme.

Table 1. Test images

Image	Size	Description
lena	512×512	lady, USC image database
peppers	512×512	vegetables, USC image database
bull	1024×768	computer-generated bull
glasses	1024×768	ray-traced glasses/dishes
wheel	512×512	antialiased color wheel

The remainder of this paper is organized as follows. First, Section 2 introduces our mesh model for images. Section 3 presents our TA mesh-generation method. Then, in Section 4, through experimental results, we evaluate the performance of our method, showing it to yield much better image approximations than the ED scheme. Finally, Section 5 concludes the paper with a summary of our work.

In our work, we have employed numerous 8-bit/sample grayscale (lattice-sampled) images as test data. Herein, we present results for a small representative subset of these images, namely the subset listed in Table 1. This subset was deliberately chosen to include both photographic and computer-generated imagery.

2. Mesh Model for Images

To begin, we introduce the mesh model for images employed by our proposed approach. The mesh model used in our work is based on constrained Delaunay triangulations [1]. In what follows, let $\mathbb{Z}$ denote the set of integers. Consider an image ϕ defined at the points $\Lambda = \{0, 1, \ldots, W-1\} \times \{0, 1, \ldots, H-1\}$ (i.e., an image sampled on a rectangular grid of width W and height H). Let $\Gamma = [0, W-1] \times [0, H-1]$. A mesh model for ϕ is completely characterized by: 1) a set $P = \{p_i\}$ of sample

*This work was supported by the Natural Sciences and Engineering Research Council of Canada.

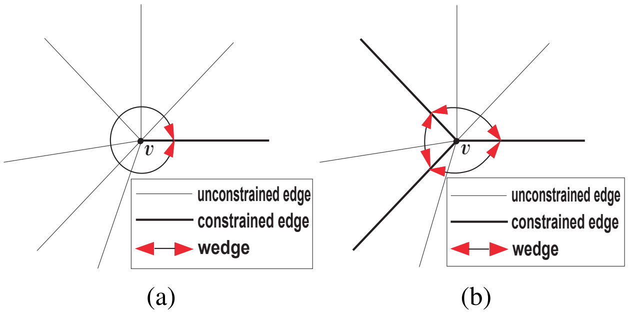

Figure 1. The relationship between vertices, constrained edges, and wedges. The (a) single wedge, and (b) multiple-wedge cases.

points, where $p_i = (x_i, y_i) \in \frac{1}{2}\mathbb{Z}^2 \cap \Gamma$; 2) a set E of edge constraints (i.e., a set of pairs of sample points from P); and 3) for each sample point p_i, one or more wedge values (where the term "wedge value" will be defined precisely later). The quantities P and E along with the associated wedge values are used to determine a function $\hat{\phi}$ defined on Γ, where $\hat{\phi}$ is an approximation of ϕ. From the mesh model $\hat{\phi}$, a lattice-sampled image can be reconstructed by straightforward rasterization algorithms (such as four-times supersampling). As a matter of terminology, we refer to the quantity $|P| / |\Lambda|$ as the **sampling density**. In what follows, we explain how $\hat{\phi}$ is defined in terms of P, E, and the wedge values.

First, we construct a constrained Delaunay triangulation of P with the constrained edges E, which serves to partition the image domain Γ into triangle faces. The constrained edges are chosen to correspond to image edges. For each vertex $v \in P$, the set of faces incident on v is partitioned into what are called wedges. In particular, a **wedge** is a set of consecutive faces in a loop around a vertex v that are not separated by any constrained edge. This definition is illustrated in Figure 1. If zero or one constrained edge is incident on the vertex v, all faces incident on v form a single wedge, as shown in Figure 1(a). Otherwise, if n constrained edges are incident on v (where $n \geq 2$), the faces incident on v form n wedges, as shown in Figure 1(b). Wedges are used to facilitate the modelling of discontinuities (i.e., image edges). Since constrained edges are chosen to correspond to image edges, a vertex $v \in P$ that has more than one wedge must be located along a discontinuity (i.e., image edge). Each wedge of a vertex has associated with it what is called a wedge value. The **wedge value** z of the wedge w belonging to vertex v specifies the limit of $\hat{\phi}(p)$ as p approaches v from points inside the wedge w.

Now, we specify precisely how the function $\hat{\phi}$ is defined at each point $p \in \Gamma$. There are two cases to consider: 1) p is not on a constrained edge; 2) p is on a constrained edge.

Case 1. First, let us consider the case that p is not on a constrained edge. Let f denote a face of the triangulation with vertices $p_i = (x_i, y_i)$, $p_j = (x_j, y_j)$, and $p_k = (x_k, y_k)$ that contains the point p. Let z_i, z_j, and z_k denote the wedge values for the face f corresponding to the vertices p_i, p_j, and p_k, respectively. Then, $\hat{\phi}(p) = g(p)$, where the function g is the unique planar interpolant that passes through the points (x_i, y_i, z_i), (x_j, y_j, z_j), and (x_k, y_k, z_k).

Case 2. Next, let us consider the case that p is on a constrained edge. If p is not an endpoint of a constrained edge, $\hat{\phi}(p)$ is the average of the values on the two sides of the image discontinuity (computed as in case 1). On the other hand, if p is an endpoint of a constrained edge (i.e., a vertex in the triangulation), $\hat{\phi}(p)$ is the average of all wedge values for (the vertex) p.

3. Proposed Mesh-Generation Method

In order for the mesh model introduced in the previous section to be useful, we need a means to choose the model parameters (i.e., P, E, and the wedge values), given an arbitrary input image ϕ. In particular, we want to be able to choose the model parameters such that $|P| = K$, where K is the target (i.e., desired) number of sample points for the mesh. Recall that P consists of triangulation vertices, and each vertex can have one or more wedge values. As a matter of terminology, a vertex with exactly one wedge is called a **nonedge point**, while a vertex with more than one wedge is called an **edge point**. To construct P and E, we first choose a set P_e of edge points along with the corresponding edge constraints E. Next, we choose a set P_n of nonedge points. Then, P is chosen simply as $P = P_e \cup P_n$. In practical terms, the sample points in P_e allow for a good image approximation around image edges, while the sample points P_n permit a good approximation away from image edges. Since we require that $|P| = K$, P_e and P_n are chosen subject to the constraint that $|P_e| + |P_n| = K$. So, in summary, our proposed method involves the following three steps: 1) Select P_e and E. 2) Choose P_n, and let $P = P_e \cup P_n$. 3) Select the wedge values for each vertex in P. In what follows, we describe each of these three steps in more detail.

SELECTION OF EDGE POINTS P_e AND EDGE CONSTRAINTS E. To choose P_e and E, we first need to locate and represent the image edges. To this end, we employ an edge-detection algorithm to produce the (binary) edge map that locates the discontinuities in the image. To detect edges, the modified Canny edge detector proposed in [2] is employed. This edge detector has low and high thresholds, which we denote as τ_{low} and τ_{high}, respectively. For simplicity, we employ only a single threshold parameter τ and then choose τ_{low} and τ_{high} in terms of τ as given by $\tau_{low} = \tau$ and $\tau_{high} = 2\tau$. Since the accuracy of the detected edges significantly impacts the quality of the mesh produced, we locate edges with half-pixel accuracy. This is done by applying the edge detector to a higher resolu-

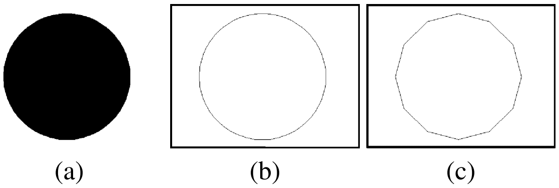

Figure 2. Process of producing simplified polylines. (a) Original image, (b) edge map, and (c) simplified polylines.

tion version of the image produced by linear interpolation, which is sampled on the grid $\frac{1}{2}\mathbb{Z}^2 \cap \Gamma$. After the edge map has been obtained, each group of edge pixels in the edge map that are 8-connected are joined together to form a polyline. In cases where a polyline has one or more self-intersections (excluding loops), the polyline is split at each intersection point. In this manner, the final set of polylines obtained are guaranteed not to have any self-intersections (although loops are permitted).

Having generated the polylines (which correspond to image edges), we reduce the number of the points in each of the polylines by using the Douglas-Peucker (DP) polyline-simplification algorithm [3]. The DP algorithm, in effect, removes points from a polyline such that the resulting simplified polyline approximates the original to within a specified tolerance. Herein, we denote the tolerance used by the DP scheme as ε. After all of the polylines have been simplified, we choose P_e as the vertices of the simplified polylines and E as the line segments of these polylines. The process of producing simplified polylines from an image is illustrated in Figure 2.

Selection of nonedge points P_n. Having determined P_e and E, we now need to select the set P_n of nonedge points. In order to ensure that the constraint $|P| = K$ is satisfied, we need to choose P_n to be disjoint from P_e and to have $K_n = K - |P_e|$ points. To select these K_n points, we employ the ED method of Yang et al. [5]. The ED method uses error diffusion in order to select a good nonuniform sampling pattern (i.e., set of sample points) for an image. With the ED method, the number of sample points is indirectly controlled by an error diffusion threshold, which we denote by ρ herein. Moreover, in the ED method, the sensitivity of sample-point selection to local image structure (such as edges) can be controlled by a contrast parameter, which we denote by γ herein. Unfortunately, due to a startup effect in error diffusion, the ED method has a tendency, when K_n is small, to select relatively few sample points in the first few rows of an image. This can lead to unusually high approximation error. To combat this problem, we always force a small number of points from the first row of the image to be included in P_n. Also, we wish to avoid choosing points for P_n that have already been chosen for P_e. As it turns out, the γ parameter can be used to to reduce the likelihood of choosing points (via the ED method) that have already been included in the set P_e. Since the ED method may not yield exactly the desired number of points (namely, K_n), we repeatedly apply the ED algorithm, adjusting the threshold ρ until the desired number of points are obtained. With P_e and P_n determined, we now choose P as $P = P_e \cup P_n$.

Selection of wedge values. Having chosen P and E, we now construct the constrained Delaunay triangulation of P with the edge constraints E. The triangulation process partitions the image domain Γ into triangle faces, with the triangle faces incident on each vertex being grouped into wedges. For each wedge w of each vertex $v \in P$, we must now compute the corresponding wedge value z. There are two cases to consider: 1) v has exactly one wedge; 2) v has more than one wedge. Consider the case that v has exactly one wedge. In this case, there is only one wedge value z to compute, which is done as follows. If $v \in \mathbb{Z}^2$, $z = \phi(v)$; otherwise, a local linear interpolant g is constructed about v, and $z = g(v)$. Now, consider the case that v has more than one wedge. Let $v' = v + d$ where d is a displacement of length 1.5 away from v along the line that bisects the wedge. If $v' \in \mathbb{Z}^2$, $z = \phi(v)$; otherwise, a local linear interpolant g is constructed about v', and $z = g(v')$.

Choice of γ, τ, and ε. As seen above, our mesh-generation method has three parameters, namely, γ, τ, and ε. As it turns out, the choice of these parameters can greatly affect the quality of the meshes produced by our method and must be carefully chosen. Here, we provide a recommendation as to how these three parameters should be selected. Based on experimentation, we found that the contrast parameter γ is best chosen as $\gamma = 0.5$. Larger values of γ were found to place too many sample points close to edges, which is undesirable in our scheme as we already choose sample points along edges by other means. On the other hand, smaller values of γ were found to spread points too uniformly, which is also not desirable. As for the parameters τ and ε, the best choice was found to be dependent on the sampling density. Again, based on experimental results, we recommend that ε and τ be chosen as indicated in Table 2 (where the choice is dependent on the sampling density). Generally speaking, as τ increases, fewer edge points are selected. Also, as ε is increased, polylines are simplified to a greater extent, resulting in fewer edge points being chosen. If either too many or too few edge points are selected, the performance of our method will be degraded. With the choices of ε and τ given in the table, we strike the best balance for the number of edge points. In the remainder of this paper, we refer to our mesh-generation method, with the choice of γ, ε, and τ recommended above, as the **Tu-Adams (TA)** method.

Remark on nonedge point selection. As explained earlier, our proposed TA method uses the ED scheme internally in order to select nonedge points. Here,

Table 2. Recommended choice of ε and τ (which depends on sampling density)

Samp. density (%)	ε	τ
$[0, 0.7)$	2	90
$[0.7, 1.5)$	2	70
$[1.5, 2.5)$	1	50
$[2.5, 5)$	1	40

Table 3. Comparison of mesh quality obtained with TA and TA-Random methods in terms of PSNR

Image	Samp. density (%)	PSNR (dB)	
		TA	TA-Random
`lena`	1.0	**25.72**	24.35
	2.0	**29.04**	27.39
	3.0	**30.16**	28.66
	4.0	**30.38**	29.48
`peppers`	1.0	24.02	**24.38**
	2.0	**28.08**	27.43
	3.0	**29.51**	28.66
	4.0	**30.07**	29.35

we present some experimental results to show the effectiveness of this choice. For comparison purposes, we consider a modified version of our TA method, called **TA-Random**, that selects nonedge points in a random fashion. For several test images and sampling densities, we used the TA and TA-Random methods to generate meshes, and then we measured the quality of the resulting image approximations in terms of PSNR. A representative subset of the results is shown in Table 3. From these results, we can see that, at sampling densities greater than 1%, the TA method clearly outperforms the TA-Random method. In other words, our use of the ED scheme for nonedge point selection is much more effective than choosing the points randomly. At very low sampling densities, such as 1% for the `peppers` image, random point selection sometimes performs better, but at such low rates the image quality is so poor as to not be useful. So, in short, at sampling densities that are sufficiently high to be of practical interest, the use of the ED scheme for selecting nonedge points is far superior to choosing such points randomly.

4. Evaluation of Proposed Method

Having introduced our mesh model and TA mesh-generation method, we now evaluate our method's performance by comparing the quality of the mesh-based image approximations produced by our method with those generated by the ED scheme. For numerous images and sampling densities, the TA and ED methods were used to generate meshes and then the quality of the associated image approximations were measured in terms of PSNR. A representative subset of the results (for the images from Table 1) is shown in Table 4.

Table 4. Comparison of mesh quality obtained with TA and ED methods in terms of PSNR

Image	Samp. density (%)	PSNR (dB)	
		TA	ED
`lena`	1.0	**25.29**	21.67
	2.0	**29.04**	26.25
	3.0	**30.16**	28.50
	4.0	**30.38**	29.67
`peppers`	1.0	**24.50**	21.69
	2.0	**28.08**	26.63
	3.0	**29.51**	28.79
	4.0	**30.07**	29.82
`bull`	0.1	**17.06**	13.96
	0.25	**30.23**	17.60
	0.5	**34.33**	27.57
	1.0	**36.97**	34.00
`glasses`	1.0	**24.82**	20.67
	2.0	**27.98**	25.52
	3.0	**29.21**	27.98
	4.0	**29.63**	29.31
`wheel`	0.1	**25.27**	9.16
	0.25	**31.10**	12.29
	0.5	**34.19**	14.95
	1.0	**35.60**	22.36

From the results in the table, we can see that our TA method can produce meshes of much higher quality than the ED scheme in terms of PSNR. For complex images like `lena`, `peppers` and `glasses`, our TA method outperforms that ED scheme by 0.25 dB to 4.15 dB, which is quite significant. For images that have many sharp edges like `bull` and `wheel`, the margin by which our TA method beats the ED scheme is even greater, ranging from 2.97 dB to 19.24 dB.

The reconstructed `lena` images and `wheel` images produced by different methods and their corresponding triangulations are shown in Figures 3 and 4, respectively. From these figures we can see that, our TA method outperforms not only in terms of PSNR, but also in terms of the visual quality. At a very low sample density (0.25%), the reconstructed wheel image produced by our TA method achieves a significant improvement compared with that produced by

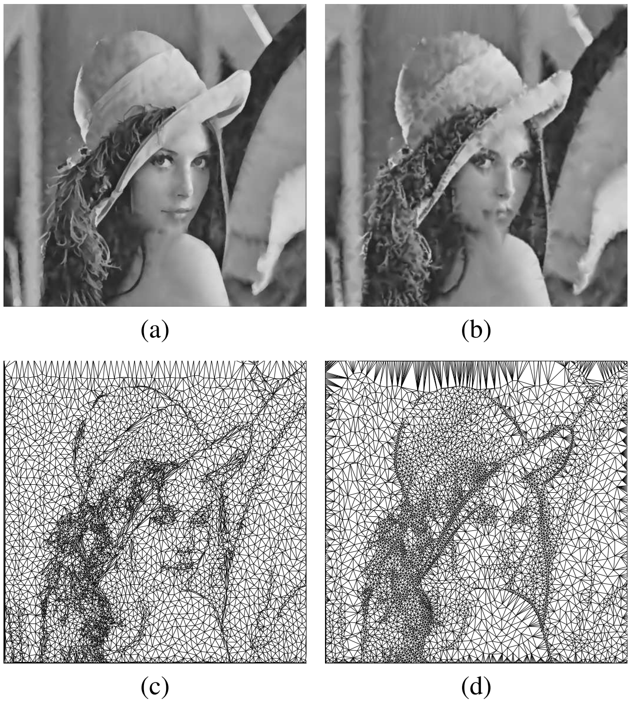

Figure 3. Image approximations obtained with the (a) TA (29.04 dB) and (b) ED (26.25 dB) methods and their corresponding triangulations (c) TA and (d) ED for the `lena` image at a sampling density of 2%.

the ED method. Observe that, with the results produced by our TA method, the edges of the triangulation align very well with the image edges, and the number of sliver triangles is much smaller, as compared to the ED scheme. Since the discontinuities are explicitly preserved with our TA method, the reconstructed images produced with our method look much sharper and clearer, and have much higher PSNR than those generated by the ED scheme. In passing, we make one last observation. Earlier, we mentioned that the ED method can sometimes place an abnormally small number of points in the first few rows of an image. This effect is evident in the top part of the triangulation in Figure 3(d).

5. Conclusions

In this paper, we have introduced a mesh model for images, which explicitly represents discontinuities (i.e., image edges), and proposed a corresponding mesh-generation method for selecting the model parameters for a given input image. Our proposed approach was shown to produce better quality mesh-based image approximations, in terms of both PSNR and subjective quality, than the previously-proposed state-of-the-art ED scheme, especially for images with sharp edges. Through the explicit representation of image discontinuities, our approach is able to perform exceedingly well, further improving upon the the state of the art in mesh-based image representations.

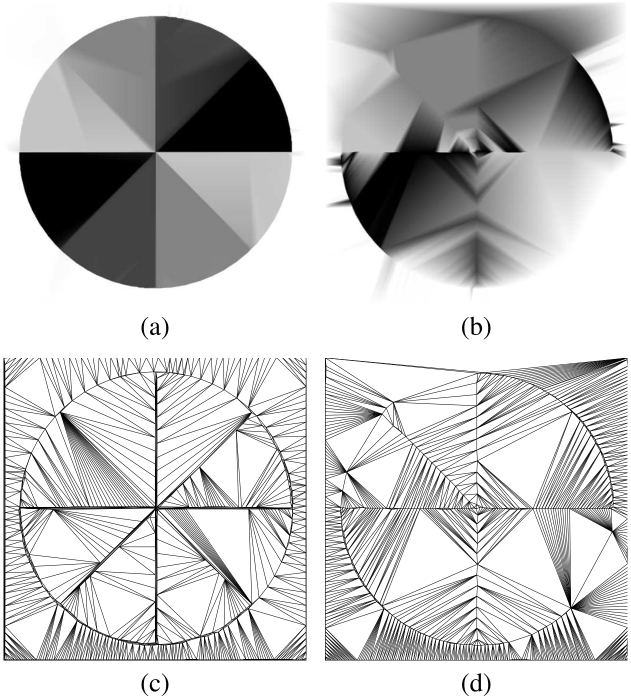

Figure 4. Image approximations obtained with the (a) TA (31.10 dB) and (b) ED (12.29 dB) methods and their corresponding triangulations (c) TA and (d) ED for the `wheel` image at a sampling density of 0.25%.

References

[1] L. P. Chew. Constrained Delaunay triangulations. *Algorithmica*, 4:97–108, 1989.

[2] L. Ding and A. Goshtasby. On the canny edge detector. *Pattern Recognition*, 34(3):721–725, March 2001.

[3] D. Douglas and T. Peucker. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Cartographica: The International Journal for Geographic Information and Geovisualization*, 10(2):112–122, October 1973.

[4] M. A. Garcia, B. Vintimilla, and A. Sappa. Approximation and processing of intensity images with discontinuity-preserving adaptive triangular meshes. Sixth European Conference on Computer Vision, LNCS Vol. 1842, Springer Verlag, Dublin, Ireland, July 2000.

[5] 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.