

# A Fast Algorithm for Accurate Content-Adaptive Mesh Generation\*

Yongyi Yang, Miles N. Wernick, and Jovan G. Brankov

Dept. of Electrical and Computer Engineering  
Illinois Institute of Technology  
3301 S. Dearborn Street, Chicago, Illinois 60616 USA

## Abstract

*In a previous paper [1] we proposed a computationally efficient approach for content-adaptive mesh generation used for image representation. In this paper we provide a theoretical basis for that method, which leads to an improved version of the algorithm. An error bound is derived for a mesh representation of an image based on the theory of function interpolation. From this result a more-accurate scheme is proposed for placement of mesh elements in the image domain according to the image content. Experimental results, comparing to other methods, show that a highly accurate image representation can be obtained at extremely low computational cost by the proposed technique.*

## 1 Introduction

In recent years, mesh modeling of images has found several applications in image processing, including image compression, motion tracking and compensation, and medical image analysis (see, for example, [2–6]). Mesh modeling of an image involves partitioning the domain of the image into a collection of non-overlapping (generally polygonal) patches, called *mesh elements*, then describing the intensity over each element through interpolation. Mesh modeling provides an efficient and compact representation of an image and, more importantly, is an effective tool for tracking rigid and non-rigid motion in image sequences.

A critical issue in mesh modeling is how to determine the best mesh structure for a given image. Several approaches for mesh generation have been proposed. One approach is to begin with an initial model of the image (such as a coarse regular mesh), then to refine the model in a hierarchical manner in order to reduce the approximation error [7–9]. Other approaches include physics-based modeling [10], and global mesh optimization [5].

In our initial work in [1] we proposed a new, computationally efficient approach for content-adaptive mesh generation. Our approach views the mesh model as a representation of the image using nonuniform sampling [11]. Our algorithm aims to create a mesh for which the local density of image samples (mesh nodes) is related to the local spatial-frequency content of the image. The algorithm consisted of the following steps: 1) generate a feature-map which highlights the high-frequency features in the image; 2) apply a classical error-diffusion algorithm, the Floyd-Steinberg method [12], to place the image samples (i.e., the mesh nodes) based on this feature map; and 3) use Delaunay triangulation [13] to compute the mesh structure. This algorithm is fast, non-iterative, easy to implement, and proved to be very accurate.

In our previous paper on this approach [1], it was presented as an *ad hoc* procedure. In this paper we provide a theoretical basis for the concept, and derive a result on the error bound of a mesh model based on the theory of function interpolation. This result leads to a modification to our earlier method that greatly increases the accuracy of the approximation. The new method is more accurate than other methods we have compared it to, and retains the major advantage of the previous approach which is its extremely low computational cost.

## 2 Theoretical Basis

### 2.1 Mesh Representation

Let  $f(\mathbf{x})$  denote an image function defined over a domain  $D$ , which has been divided into a total of  $M$  non-overlapping mesh elements  $D_m, m = 1, 2, \dots, M$ . Assuming that each element  $D_m$  has  $N$  nodes, the image function is represented over each  $D_m$  as follows

$$\hat{f}(\mathbf{x}) = \sum_{n=1}^N f(\mathbf{x}_n) \varphi_{m,n}(\mathbf{x}), \quad \text{for } \mathbf{x} \in D_m, \quad (1)$$

where  $\varphi_{m,n}(\mathbf{x})$  is the interpolation basis function associated with the  $n$ th node  $\mathbf{x}_n$  of  $D_m$ . In practice, the

\*This research was supported by NIH/NHLBI Grant HL65425 and the Whitaker Foundation

elements  $D_m$  are often chosen to be triangles or quadrangles because of the geometrical simplicity of these shapes. In this study triangular elements are used.

The mesh representation in Eq. (1) assumes a form of signal representation based on non-uniform sampling. By definition, each basis function  $\varphi_{m,n}(\mathbf{x})$  in Eq. (1) has support only over its associated element  $D_m$ . Thus, the total contribution by a particular nodal value  $f(\mathbf{x}_n)$  to the image is strictly limited to those elements associated with  $\mathbf{x}_n$ .

For notational simplicity, it is generally assumed in this paper that the image function  $f(\mathbf{x})$  is defined over a two-dimensional (2-D) domain  $D$ . When desired, the image function  $f(\mathbf{x})$  may be explicitly written as  $f(x, y)$  with the understanding that  $\mathbf{x} = (x, y)$ . It is noted, however, that the rest of the development can also be extended directly to a higher dimensional case such as 3-D.

## 2.2 Error Analysis

A key question in mesh representation is approximation error. Error bounds have been derived in finite element analysis [14] where the goal is to provide algorithmic convergence analysis. Here, we derive a result for the purpose of adaptive mesh generation. For brevity we simply state the result without proof.

**Theorem 1** *Let  $T$  denote a triangle on the two-dimensional plane  $R^2$ , and let  $C^2[T]$  denote the space of all real-valued functions defined on  $T$  which have 2nd partial derivatives that are continuous on  $T$ . Assume that  $\hat{f}(x, y)$  is the linear interpolation of  $f(x, y)$  at the vertices of  $T$ . Then for each point  $(x, y) \in T$*

$$\left| f(x, y) - \hat{f}(x, y) \right| \leq \frac{1}{4} M_2 h^2, \quad (2)$$

where  $h$  is the length of the longest side of  $T$ , and  $M_2$  is the least upper bound on the 2nd-order directional derivative of  $f(x, y)$  over  $T$ , i.e.,

$$M_2 \triangleq \max_{(x,y) \in T} \max_{\theta \in [0, 2\pi]} |f''_{\theta}(x, y)|, \quad (3)$$

where  $f''_{\theta}(x, y)$  denotes the 2nd-order derivative of  $f(x, y)$  at point  $(x, y)$  along the unit vector  $\mathbf{u}_{\theta} = (\cos \theta, \sin \theta)^t$ .

The result in Eq.(2) provides an important theoretical basis for the development of our mesh generation algorithm to be described. First, it states that the approximation error bound is proportional to the maximum magnitude assumed by the 2nd directives of the image function over the element  $T$ . Second, it states that this error bound is also proportional to the square of the length of the longest side of  $T$ . Note that

the latter is simply proportional to the area of  $T$  provided that it is not excessively elongated (i.e.,  $T$  not having an angle too small). Based on the above observation, we argue the following: *to achieve a uniform error level throughout the image, a good mesh generation scheme should seek to place small elements (in terms of area) in image regions where the 2nd directional directive is large, and larger elements in regions where the 2nd directional directive is small.*

## 3 Content-Adaptive Mesh Generation

Based on this concept, we propose the following three-step mesh generation algorithm. First, a feature map, denoted by  $\sigma(x, y)$ , is extracted from the image  $f(x, y)$  based on the largest magnitude of its second directional directives. In the second step, the Floyd-Steinberg method is employed to distribute samples non-uniformly in the image domain, with density proportional to  $\sigma(x, y)$ . These samples serve as the nodes of the mesh model. In the third and final step, a 2-D Delaunay triangulation algorithm [13] is used to connect the mesh nodes. The resulting mesh structure consists of triangular mesh elements which are automatically adapted to the content of the image. The details of these steps are further described below.

### 3.1 Image Feature-Map Extraction

For convenience, let  $g(x, y)$  denote the largest magnitude of the 2nd-order directional derivative of  $f(x, y)$  at point  $(x, y)$ , that is

$$g(x, y) = \max_{\theta \in [0, 2\pi]} |f''_{\theta}(x, y)|.$$

The feature map function  $\sigma(x, y)$  is determined as

$$\sigma(x, y) = \left( \frac{g(x, y)}{A} \right)^{\gamma}, \quad (4)$$

where  $A$  is the largest value of  $g(x, y)$ , and  $\gamma$  is a constant with  $0 < \gamma \leq 1$ . The role of  $A$  in Eq. (4) is simply to normalize the 2nd derivative magnitude  $g(x, y)$  within the range between 0 and 1, while  $\gamma$  is used to enhance weak edge features in the image.

For computation of  $g(x, y)$ , one can derive the following:

**Corollary 1** *Let  $\mathbf{H}_{(x,y)}$  denote the Hessian matrix of  $f(x, y)$  at  $(x, y)$ , and let  $\lambda_{1,2}(x, y)$  denote the two eigenvalues of  $\mathbf{H}_{(x,y)}$ . Then we have*

$$g(x, y) = \max \{ |\lambda_1(x, y)|, |\lambda_2(x, y)| \}. \quad (5)$$

Note that the eigenvalues of  $\mathbf{H}_{(x,y)}$  are given by

$$\lambda_{1,2}(x, y) = \frac{\frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}}{2} \pm \sqrt{\left( \frac{\frac{\partial^2 f}{\partial x^2} - \frac{\partial^2 f}{\partial y^2}}{2} \right)^2 + \left( \frac{\partial^2 f}{\partial x \partial y} \right)^2}$$

where for simplicity the arguments of  $f(x, y)$  have been dropped.

Furthermore, one can show the following:

**Corollary 2** Let  $M(x, y)$  denote the largest magnitude of the 2nd partial derivatives of  $f(x, y)$  at point  $(x, y)$ , i.e.,

$$M(x, y) \triangleq \max \left\{ \left| \frac{\partial^2 f(x, y)}{\partial x^2} \right|, \left| \frac{\partial^2 f(x, y)}{\partial x \partial y} \right|, \left| \frac{\partial^2 f(x, y)}{\partial y^2} \right| \right\}. \quad (6)$$

Then  $g(x, y) \leq 2M(x, y)$ .

Interestingly, by the definition of  $g(x, y)$  we also have

$$g(x, y) \geq \max \left\{ \left| \frac{\partial^2 f(x, y)}{\partial x^2} \right|, \left| \frac{\partial^2 f(x, y)}{\partial y^2} \right| \right\}.$$

Clearly,  $M(x, y)$  is relatively easier to compute when compared to  $g(x, y)$ . To simplify the computation of the feature map, we may simply use  $M(x, y)$  in place of  $g(x, y)$  in Eq. (4).

### 3.2 Adaptive Mesh-Node Generation

The Floyd-Steinberg algorithm [12] is a classical error-diffusion algorithm widely used in digital halftoning, where the spatial density of ink dots is used to represent the image intensity. In our application, the halftoning algorithm places the mesh nodes, i.e., image samples  $\mathbf{x}_n$  in Eq. (1), in accordance with the density specified by the feature map  $\sigma(x, y)$ .

### 3.3 Delaunay Triangulation

Delaunay triangulation connects a given set of mesh nodes in such a way that the circle circumscribing any triangular element contains only the nodal points belonging to that triangle (except in the case where four or more nodal points are co-circular). Delaunay triangulation can yield a well-structured mesh at a reasonable computational cost. Most importantly, the use of Delaunay triangulation in our case can avoid producing excessively elongated elements, thereby further reducing the error bound in Eq.(2).

## 4 Numerical Results

Shown in Fig. 1(a) is a  $128 \times 128$  section cropped from the original image ‘‘Lena’’; Fig. 1(b) is the feature map image  $\sigma(x, y)$ . For computing  $\sigma(x, y)$ ,  $\gamma = 3/4$  and  $M(x, y)$  rather than  $g(x, y)$  were used in Eq. (4). Shown in Fig. 1(c) is the mesh structure obtained using the proposed algorithm, in which only 3 348 mesh nodes (about 20% of the number of pixels) were used. Note that dense mesh elements have been automatically placed in regions containing high-frequency features (such as edges), while coarse elements have been

placed in relatively flat areas of the image. The image reconstructed from this mesh by Eq. (1) using linear interpolation is shown in Fig. 1(d). Figures 1(e) and (f) shows the mesh structure and reconstructed image obtained using the procedure described in our previous work [1] for the same image with the same number of mesh nodes. Similar results are given in Figs. 1(g) and (h) for the well-known quadtree mesh generation method [5].

To evaluate the interpolated images we used the peak-signal-to-noise ratio (PSNR) defined as:

$$\text{PSNR} = 10 \log \left( M \times N \times 255^2 / \|\hat{\mathbf{f}} - \mathbf{f}\|^2 \right), \quad (7)$$

where  $\hat{\mathbf{f}}$  and  $\mathbf{f}$  denote the interpolated and original images, respectively, and  $M \times N$  is the image dimension. The PSNR values for the methods considered are shown in the caption of Fig. 1. The mesh representation generated from our new algorithm exhibits the best accuracy, both visually and quantitatively, for this image.

The optimization approach described in [5] was next applied to further optimize the mesh structures obtained by each of the three methods, which, of course dramatically increases the computational cost. The resulting PSNR values are: 30.48 dB for our new method, 28.70 dB for our old method, and 28.21 dB for the quadtree method. Interestingly, the new mesh representation in Fig. 1(c) *without optimization* offers the better accuracy than the other methods produce *with optimization*. Thus, the new method produces not only the best images, but does so at extremely low computational cost.

Additional results indicate that the new method consistently outperforms both the quadtree mesh-generation algorithm with optimization and our previous approach. Due to space limitation the details of these experiments are not included here, but will be presented at the conference.

Currently we are applying the proposed technique in medical image reconstruction for cardiac image sequences, where fast, compact image representation can greatly improve the reconstruction speed and image quality. These results are presented in a separate paper in this conference [15].

## References

- [1] J. Lee, Y. Yang, and M. N. Wernick, ‘‘A new approach for image-content adaptive mesh generation,’’ *IEEE Int. Conf. Image Proc.*, Vancouver, Sept., 2000.
- [2] K. Aizawa, and T. S. Huang, ‘‘Model-based image coding: advanced video coding techniques for very

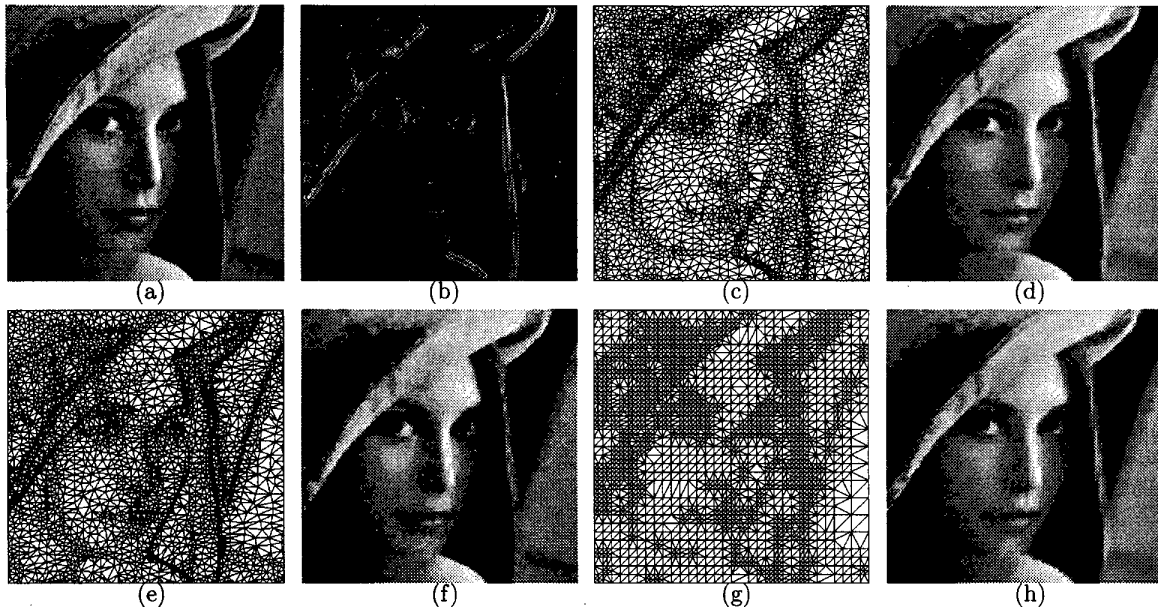


Figure 1: (a) A  $128 \times 128$  section of original image “Lena”; (b) Feature map extracted from the image in (a); (c) Mesh structure obtained using the proposed algorithm, 3 348 mesh nodes used; (d) Image represented using the mesh in (c), PSNR=30.07 dB; (e) Mesh obtained from previous method [1], same number of mesh nodes as in (c) used; (f) Image represented using the mesh in (e), PSNR=27.61 dB; (g) Mesh obtained using the quadtree method, same number of mesh nodes as in (c) used; (h) Image represented using the mesh in (e), PSNR=27.22 dB.

- low bit-rate applications,” *Proc. of IEEE*, vol. 83, no.2, pp. 259–271, Feb. 1995.
- [3] F. Davoine, M. Antonini, J. Chassery, and M. Barlaud, “Fractal image compression based on Delaunay triangulation and vector quantization,” *IEEE Trans. Image Proc.*, vol. 5, no. 2, pp. 338–346, 1996.
- [4] H. Benoit-Cattin, P. Joachimsmann, A. Planat, S. Valette, A. Baskurt, R. Prost, “Active mesh texture coding based on warping and DCT,” *IEEE Int. Conf. Image Proc.*, Kobe, Japan, 1999.
- [5] Y. Wang and O. Lee, “Active mesh—a feature seeking and tracking image sequence representation scheme,” *IEEE Trans. Imag. Proc.*, vol. 3, no. 5, 1994.
- [6] Y. Altunbasak and A. M. Tekalp, “Closed-form connectivity-preserving solutions for motion compensation using 2-D meshes,” *IEEE Trans. Imag. Proc.*, vol. 6, no. 9, pp. 1255–1269, 1997.
- [7] C. L. Huang, and C. Y. Hsu, “A new motion compensation method for image sequence coding using hierarchical grid interpolation,” *IEEE Trans. Circuits Syst. Video Tech.*, vol. 4, pp.44–51, 1994.
- [8] T. Gevers and A. W. Smeulders, “Combining region splitting and edge detection through guided Delaunay image subdivision,” *IEEE Int. Conf. Comp. Vision, Pattern Recog.*, pp.1021–1026, Puerto Rico, June 1997.
- [9] M. A. Garcia, B. X. Vintimilla, and A. D. Sappa, “Efficient approximation of gray-scale image through bounded error triangular meshes,” *IEEE Int. Conf. Image Proc.*, Kobe, Japan, 1999.
- [10] D. Terzopoulos and M. Vasilescu, “Sampling and reconstruction with adaptive meshes,” *IEEE Int. Conf. Comp. Vision, Pattern Recog.*, pp. 829–831, 1992.
- [11] R. J. Marks II, ed., *Advanced Topics in Shannon Sampling and Interpolation Theory*, Springer-Verlag, 1993.
- [12] R. Floyd and L. Steinberg, “An adaptive algorithm for spatial gray scale,” *SID Int. Sym. Digest of Tech. Papers*, pp. 36–37, 1975.
- [13] F. Preparata and M. Shamos, *Computational Geometry—An Introduction*, Springer-Verlag, New York, 1985.
- [14] P. M. Prenter, *Splines and Variational Methods*, John Wiley & Sons, 1975.
- [15] J. G. Brankov, Y. Yang, and M. N. Wernick, “Tomographic image reconstruction using content-adaptive mesh modeling,” *IEEE Int. Conf. Image Proc.*, Thessaloniki, Greece, Oct. 2001.