ICIP-2002 Paper Proposal:

ACCURATE MESH REPRESENTATION OF VECTOR-VALUED (COLOR) IMAGES

Jovan G. Brankov, Yongyi Yang, and Miles N. Wernick
Dept. of Electrical and Computer Engineering
Illinois Institute of Technology
3301 S. Dearborn Street, Chicago, IL 60616

ABSTRACT
In this work we present a fast procedure for content-adaptive mesh representation of vector-valued (e.g., color) images. The goal is to obtain a single mesh structure that accurately represents all the individual components of the image. The proposed method is justified by an error bound that is rigorously derived for such a representation. It employs an error-diffusion type algorithm to place the mesh nodes non-uniformly in the image domain according to the image content. Experimental results demonstrate that: (1) a compact and accurate representation for color images can be achieved at low computational cost by the proposed algorithm; and (2) joint treatment of the different image components by the proposed algorithm can result in a more accurate mesh representation than a mesh based on a single image component (such as intensity) alone.

1. INTRODUCTION

Mesh modeling of an image involves partitioning the image domain into a collection of non-overlapping (generally polygonal) patches, called mesh elements. The image function is then determined over each element through interpolation. Mesh modeling provides an efficient and compact representation of an image and is an effective tool for tracking rigid and non-rigid motion in image sequences. As a result, mesh modeling of images has found many important applications in recent years in image processing, including, for example, image compression [1-4], motion tracking and compensation [5-9], image processing through geometric manipulation [10], medical image analysis [11], and more recently, tomographic image reconstruction [12].

A critical issue in mesh modeling is how to determine the mesh structure in a mesh model for a given image. Several approaches for mesh generation have been proposed in image processing, almost all explicitly for scalar-valued images. One approach is to begin with an initial model of the image (such as a coarse, regular mesh), then to successively refine the model in a hierarchical manner in order to reduce the approximation error [10,13]. Other approaches include physics-based modeling [14], and global mesh optimization [5].

The purpose of this work is to study a method that can generate a compact and accurate mesh representation of vector-valued images. A vector-valued image is defined here as a signal that consists of two or more components defined over a common 2D domain. Examples of vector-valued images include color images, multi-spectral images, and multi-modality medical images (e.g., CT/MRI). Our goal is to obtain a mesh structure for a vector-valued image so that all the individual components of the image can be accurately represented by the same mesh.

Toward this goal, an error bound, defined jointly for all the different components of a vector-valued image, is derived for a vector-valued mesh representation. Based on this result, we design an algorithm which aims to adaptively distribute mesh nodes (hence mesh elements) in the image domain in such a way that the error level by the mesh representation is kept small over individual elements.

Specifically, the algorithm consists of the following three steps: 1) generate a feature map which highlights the spatial distribution of the high-frequency features in all the image components; 2) apply an error-diffusion algorithm—based on the well-known Floyd-Steinberg algorithm [15]—to distribute the mesh nodes in the image domain; and 3) use Delaunay triangulation [16] to compute the mesh structure. This approach is fast, non-iterative, easy to implement, and has proven to be very accurate.

The proposed approach is an extension of our previous work in [17], wherein mesh representation was studied for scalar intensity images. It was demonstrated that this approach, compared to several existing methods, can produce a more accurate content-adaptive mesh representation for an image at very low computational cost.

We point out that in the literature there exists work for compression of color images for the purpose of efficient storage and transmission. While a compact, accurate mesh model of a vector-valued image may be applicable for
compression, our study is intended principally for model-based image processing tasks. Our ultimate goal is to apply this model to tomographic reconstruction of dual-modality medical images. Our previous work [12] demonstrated that such an approach (with intensity images) can outperform several well-known reconstruction algorithms.

2. THEORETICAL BASIS

2.1. Mesh Representation

Let \( \mathbf{F}(\mathbf{p}) = (f_1(\mathbf{p}), f_2(\mathbf{p}), \ldots, f_K(\mathbf{p}))^T \) denote a vector-valued image function defined over a domain \( D \subset \mathbb{R}^2 \), where \( f_k(\mathbf{p}) \) denotes its \( k \)th component, \( k = 1, 2, \ldots, K \), and \( \mathbf{p} = (x, y) \). Let \( \mathbf{\hat{F}}(\mathbf{p}) \) denote the representation of \( \mathbf{F}(\mathbf{p}) \) using a common mesh, where the domain \( D \) is partitioned into a number, say \( M \), of non-overlapping mesh elements, which are denoted by \( D_m \), \( m = 1, 2, \ldots, M \). Then over each element \( D_m \) each component \( \hat{f}_k(\mathbf{p}) \) of \( \mathbf{\hat{F}}(\mathbf{p}) \) is given by

\[
\hat{f}_k(\mathbf{p}) = \sum_{n=1}^{N} f_k(\mathbf{p}_n) \varphi_{m,n}(\mathbf{p}), \quad \mathbf{p} \in D_m, \tag{1}
\]

where \( \varphi_{m,n}(\mathbf{p}) \) is the interpolation basis function associated with the \( n \)th node \( \mathbf{p}_n \) of \( D_m \), and \( N \) is the total number of mesh nodes used. Note that in (1) the same set of basis functions is used for all the different image components.

The mesh representation in Eq. (1) assumes a form of signal representation based on non-uniform sampling, where the mesh nodes serve as image samples. In this study, triangular elements are used for \( D_m \), so \( N = 3 \). Also, linear interpolation functions are used for \( \varphi_{m,n}(\mathbf{p}) \).

2.2. Error Analysis

The key to the mesh representation in (1) is its accuracy. For a given number of mesh nodes, our goal is to obtain a common mesh structure, defined by the mesh elements \( D_m \), so that it provides a compact, accurate representation of all the image components.

Let us introduce an error metric between \( \mathbf{F}(\mathbf{p}) \) and its mesh representation \( \mathbf{\hat{F}}(\mathbf{p}) \). For generality, define the error at each \( \mathbf{p} \in D_m \) using the \( p \)-norm in \( \mathbb{R}^K \) as

\[
e(\mathbf{p}) = \left| \mathbf{F}(\mathbf{p}) - \mathbf{\hat{F}}(\mathbf{p}) \right|_p = \left[ \sum_{k=1}^{K} (f_k(\mathbf{p}) - \hat{f}_k(\mathbf{p}))^p \right]^{1/p}. \tag{2}
\]

Then the following result can be derived (the proof is omitted for brevity):

Theorem 1. Let \( T \) denote a triangle in the 2D plane \( \mathbb{R}^2 \), and let \( M_{2,k} \) denote the least upper bound on the magnitude of the 2nd order directional derivative of \( f_k(\mathbf{p}) \) over \( T \), \( k = 1, 2, \ldots, K \). Assume that \( \hat{f}_k(\mathbf{p}) \) is the linear interpolation of \( f_k(\mathbf{p}) \) at the vertices of \( T \). Then for each point \( \mathbf{p} \in T \)

\[
e(\mathbf{p}) \leq \frac{1}{4} \left[ \sum_{k=1}^{K} (M_{2,k})^2 \right]^{1/2} h^2, \tag{3}
\]

where \( h \) is the length of the longest side of \( T \).

3. CONTENT-ADAPTIVE MESH GENERATION

The result in (3) provides a fundamental basis for the development of our mesh generation algorithm. It states that the approximation error bound is proportional to two factors: (1) the maximum magnitude assumed by the 2nd directional directives of the image components; and (2) the square of the length of the longest side of \( T \). Note that the latter is also proportional to the area of \( T \) provided that it is not excessively elongated. Based on this observation, we argue that a good mesh generation scheme should try to place small (in area) elements in regions where the magnitude of the 2nd directional directives of the image components is large, and conversely, larger elements should be used in regions where the magnitude of the 2nd directional directives is relatively small in order to achieve a balanced error level throughout the image domain.

We propose the following three-step mesh generation algorithm that aims specifically to achieve this goal at low computational cost: First, a feature map \( \sigma(\mathbf{p}) \) is extracted from the vector-valued image based on the largest magnitude of its second directional directives. Second, the well-known Floyd-Steinberg error-diffusion algorithm, a method designed for digital halftoning [15], is employed to distribute mesh nodes non-uniformly in the image domain, with density proportional to \( \sigma(\mathbf{p}) \). Third and finally, a Delaunay triangulation algorithm [16] is used to connect the mesh nodes. The resulting mesh consists of triangular elements which are automatically adapted to the content of the image. The details of these steps are further described below.

3.1. Feature Map Extraction

Let \( G_k(\mathbf{p}) \), \( k = 1, 2, \ldots, K \), denote the largest magnitude of the second directional derivative of \( f_k(\mathbf{p}) \) at point \( \mathbf{p} \). The feature map function is defined as

\[
\sigma(\mathbf{p}) = \left[ \sum_{k=1}^{K} (G_k(\mathbf{p}))^2 \right]^{1/2}. \tag{4}
\]

To compute \( G_k(\mathbf{p}) \), one can derive the following:
Corollary 1. Let $H_{i,p}$ denote the Hessian matrix of $f_i(p)$ at $p$, and let $\lambda_{i,j}(p), i=1,2,$ denote the eigenvalues of $H_{i,p}$. Then, we have

$$G_i(p) = \max \left\{ \left| \lambda_{i,1}(p) \right|, \left| \lambda_{i,2}(p) \right| \right\}.$$

(5)

3.2. Content-Adaptive Placement of Mesh Nodes

The classical Floyd-Steinberg error-diffusion algorithm was originally intended for digital halftoning, where the objective is to use the spatial density of ink dots to represent the image intensity. We apply it here for placing mesh nodes in accordance with the density specified by the feature map $\sigma(p)$. Besides being fast, with this algorithm we can easily control the number of mesh nodes by adjusting the threshold value of half-toning.

3.3. Delaunay Triangulation

Among its many interesting properties, Delaunay triangulation is known to yield a well-structured mesh at a reasonable computational cost. The use of Delaunay triangulation avoids producing excessively elongated elements, thereby reducing the error bound in (3).

3.4. Mesh Nodal Value

Once the mesh structure is obtained, the image can then be represented by interpolation over each element from its nodal points. The nodal value $f_i(p_n)$ in (1) can simply be taken as the image value at $p_n$. Alternatively, it can also be determined using a least-squares fit procedure such that the mean-squared-error of the interpolated image $\hat{f}_i(p), k=1,2,\cdots,K$, is minimized. The latter is found to be more accurate and is adopted in this study.

4. NUMERICAL RESULTS

In this section we present some results obtained by the proposed mesh generation method using color images. Shown in Fig. 1(a) is an original image, of size $128\times 128$. Shown in Fig. 1(b) is the resulting mesh obtained by the proposed algorithm when applied to the RGB components of this image, in which only 2,340 mesh nodes were used (about 14.2% of the number of pixels in the original image). In addition, $p=2$ was used in the error norm. Note that dense mesh elements have been automatically placed in regions containing high-frequency features (such as edges and textures), while coarse elements have been placed in relatively flat regains. The image represented by this mesh is shown in Fig. 1(c).

To quantify the accuracy of this mesh representation, we computed its peak-signal-to-noise ratio (PSNR) to be 30.2 dB. The PSNR is defined as

$$PSNR = \sum_{i=1}^{3} 10 \log \left( \frac{M \times N \cdot f_{\text{max}}^2}{\| f_i - \hat{f}_i \|_2^2} \right) \text{dB},$$

where $f_i$ and $\hat{f}_i$ denote the original $k$th color component and its mesh representation, respectively, $f_{\text{max}}$ is the image peak value, and $M \times N$ is the image dimension.

For comparison, we also show in Fig. 1(d) and Fig. 1(e) the mesh and interpolated image, respectively, obtained by the well-known quadtree method [5], using the same number of mesh nodes. The image has PSNR=28.9 dB.

Furthermore, we show in Fig. 1(f) the mesh structure obtained based on the intensity of the image alone using the proposed algorithm. When compared to the mesh structure in Fig. 1(b), it is clear that in this mesh structure the color transition boundaries between the hair and cheek area is not as well defined. As a result, the image represented by this mesh suffers from color bleeding at these color boundaries. Due to space limitation, this image and results obtained with other test images are not shown in this proposal.

The above results demonstrate that the proposed method can produce more accurate mesh representation of color images. Note that this is also achieved at very low computational cost.

Additional results are provided for better visualization at the website: http://www.ipl.iit.edu/brankov/Color.htm. These results show that the mesh structure produced by the proposed method is well adapted to the content of the color images.

5. CONCLUSIONS AND FUTURE WORK

In this paper we have proposed a fast mesh generation approach that can produce a compact and accurate mesh representation of vector-valued images. Currently we are studying application of the proposed mesh representation in tomographic reconstruction of dual-modality cardiac images. The use of a content-adaptive mesh model in this case can lead to fast and accurate reconstruction.

ACKNOWLEDGEMENT

The authors thank Dr. Brian L. Evans for his suggestion of applying our mesh generation method [17] to color images.

6. REFERENCES


Figure 1. (a) Original image (128x128); (b) Mesh structure obtained using the proposed algorithm based on the RGB components of the image in (a), 2,340 mesh nodes used; (c) Image represented by the mesh in (b), PSNR=30.2 dB; (d) Mesh structure obtained by the quadtree method, same number of mesh nodes as in (b) used; (e) Image represented by the mesh in (c), PSNR=28.9 dB; (f) Mesh structure obtained based on the intensity of the image in (a) alone. The image represented by this mesh suffers color bleeding at color transition boundaries in the image.