A Highly-Effective Approach for Generating Delaunay Mesh Models of RGB Color Images

Jun Luo and Michael D. Adams
Dept. of Elec. and Comp. Engineering, University of Victoria, Victoria, BC, V8P 5C2, Canada
junluo@uvic.ca and mdadams@ece.uvic.ca

***Abstract*—A highly effective method for generating Delaunay mesh models of RGB (i.e., red-green-blue) color images, known as CMG, is proposed. This method builds on ideas from the previously-proposed GPRFSED method of Adams for grayscale images to produce a method that can handle RGB color images. The key ideas embodied in our CMG method are Floyd-Steinberg error diffusion with improved initial-condition selection and greedy-point removal. Through experimental results, our CMG method is shown to outperform several competing methods that are based on a straightforward extension of grayscale mesh generators to color, with our method yielding meshes of vastly better quality at lower or comparable computational/memory cost.**

I. Introduction

Despite being the most straightforward and commonly used method for representing images, uniform (i.e., lattice-based) sampling is far from optimal. Due to the fact that most images are nonstationary, such sampling inevitably leads to oversampling in some regions and undersampling in others. This motivates an interest in nonuniform (i.e., content-adaptive) sampling for image representation. In nonuniform sampling, by intelligently choosing the sample points based on image content, the number of sample points used for representing the image can be greatly reduced while still maintaining good fidelity. Nonuniform sampling has shown to be useful in many applications, including pattern recognition [1] and feature detection [2] amongst others.

Many approaches to nonuniform sampling have been proposed over the years. Of these approaches, the ones based on triangle meshes have proven to be particularly effective and are the focus of our work herein. With triangle mesh models, the image domain is partitioned into triangles using a process known as triangulation, and then an interpolant is formed over each (triangle) face of the triangulation to form an approximating function. Triangle meshes are particularly well suited to capturing the geometric structure in images (i.e., edges and corners).

Many methods for generating mesh models of images have been proposed over the years. In our work, we focus only on mesh-generation methods based on Delaunay triangulations. Among them, three very popular methods are the ***error diffusion (ED) scheme*** of Yang et al. [3], the ***greedy point-removal (GPR) scheme*** of Demaret and Iske [4], and the ***GPRFSED scheme*** of Adams [5]. The ED method employs ***Floyd-Steinberg error diffusion (FSED)*** [6] to select all of the sample points of the mesh in one shot, and then triangulates them using a Delaunay triangulation. The GPR method starts with a triangulation containing all sample points from the image sampling grid as vertices (i.e., sample points), and then repeatedly removes vertices until the desired mesh size is achieved. The GPRFSED method combines the ideas of the ED and GPR methods. That is, the GPRFSED method employs the ED scheme to select the sample points for an initial mesh whose size is larger than desired, and then performs mesh simplification using the GPR scheme until the desired mesh size is achieved.

Although the ED, GPR, and GPRFSED methods are very effective, they are only capable of generating mesh models of images that are grayscale (i.e., have a single component). In many applications, however, the images involved are color. Since the above three methods are very effective, it would be highly desirable if we could find a good way to extend these methods to handle RGB (i.e., red-green-blue) color images. Obviously, we can trivially extend any mesh-generation method for grayscale images to handle RGB color images as follows. Given a color image as input, first convert the color image to grayscale using a standard RGB-to-grayscale conversion [7], and then generate a mesh for the resulting grayscale image. Then, after obtaining a mesh model for this grayscale image, replace the grayscale sample values in the generated mesh by their corresponding RGB values. In the remainder of this paper, we refer to the trivial extensions of the ED, GPRFSED, and GPR methods to color using the preceding strategy as the ***CED***, ***CGPRFSED***, and ***CGPR*** methods, respectively. As one might expect, however, this trivial solution of using the CED, CGPRFSED, and CGPR methods in order to handle color images is not the most effective. This motivated us to seek a better way to extend the ideas embodied in the ED, GPRFSED, and GPR methods to allow the handling of RGB color images.

In this paper, we propose a highly effective mesh-generation method for RGB color images known as ***CMG***, which is based on ideas from the GPRFSED scheme. Amongst other things, our CMG method includes a novel technique for reducing the undesirable start-up effects that can result from the use of classical FSED at low sampling densities. With our CMG method, by varying the initial mesh size, one can easily tradeoff between mesh quality and computational/memory cost. Through experimental results, our proposed method is shown to yield meshes of higher quality in terms of PSNR than the CED, CGPRFSED, and CGPR methods, while requiring

either lower or comparable computational/memory cost.

The remainder of this paper is organized as follows. In Section II, we present some background information related to (triangle) mesh models for image representation. In Section III, we introduce our proposed mesh-generation method, known as CMG. Then, in Section IV, we evaluate the performance of our CMG method in terms of mesh quality and computational/memory cost by comparing to the CED, CGPRFSED and CGPR methods. Finally, Section V concludes the paper with a summary of our work.

II. Background

In what follows, we introduce some background information that is essential for understanding the work presented herein. The sets of integers and real numbers are denoted as $\mathbb{Z}$ and $\mathbb{R}$, respectively. The cardinality of a set P is denoted as $|P|$. For $a, b \in \mathbb{Z}$, we define the following notation for integer ranges: $[a..b] = \{x \in \mathbb{Z} : a \leq x \leq b\}$ and $[a..b) = \{x \in \mathbb{Z} : a \leq x < b\}$. A ***triangulation*** of a finite set P of points in $\mathbb{R}^2$ is a set T of (non-degenerate) triangles satisfying the following conditions: 1) the set of all vertices of triangles in T is P; 2) the union of all triangles in T is the convex hull of P; and 3) the interiors of any two triangles in T are disjoint.

An M-component image of width W and height H is a vector-valued function ϕ defined on $D = [0, W-1] \times [0, H-1]$ and sampled on the truncated integer lattice $\Lambda = [0..W-1] \times [0..H-1]$. The ith component of the image ϕ (where $i \in [0..M)$) is denoted as ϕ_i. In the case of RGB color images (which are of primary interest herein), $M = 3$. A ***triangle mesh model*** of ϕ is characterized by: 1) a set $P = \{p_i\}_{i=0}^{|P|-1}$ of sample points, where $P \subset \Lambda$; 2) a triangulation T of P; and 3) a set $Z = \{z_i\}_{i=0}^{|P|-1}$ of function values, where $z_i = \phi(p_i)$. As a matter of terminology, we define $|P|$ and $|P|/|\Lambda|$ as the ***size*** and ***sampling density*** of the mesh model, respectively. In order to ensure that all points of Λ are covered by the triangulation T, the set P must always include all of the extreme convex hull points of the image domain.

The above mesh model is associated with a function $\hat{\phi}$ that approximates ϕ, where the ith component of $\hat{\phi}$ is denoted $\hat{\phi}_i$. Since ϕ has integer-valued components $\{\phi_i\}_{i=0}^{M-1}$, the approximating function $\hat{\phi}$ is also defined to have integer-valued components $\{\hat{\phi}_i\}_{i=0}^{M-1}$. For $i \in [0..M)$, let $\tilde{\phi}_i$ denote the function that linearly interpolates ϕ_i over each face f in T. Then, $\hat{\phi}_i$ is defined as $\hat{\phi}_i = \text{round}\left(\tilde{\phi}_i\right)$, where round denotes an operator that rounds to the nearest integer (i.e., $\hat{\phi}_i$ is simply the function $\tilde{\phi}_i$ rounded to integer values).

In our work, the particular mesh-generation problem that we address can be stated as follows: given an image ϕ and a desired number N of sample points, choose a set P of sample points where $P \subset \Lambda$ and $|P| = N$ such that the error between the original image ϕ and the approximating image $\hat{\phi}$ reconstructed from the mesh model with the sample points P is minimized. The error between ϕ and $\hat{\phi}$ is measured by the ***mean squared error (MSE)***, which is defined as $\mathsf{MSE} = \frac{1}{M|\Lambda|} \sum_{i=0}^{M-1} \sum_{p \in \Lambda} [\hat{\phi}_i(p) - \phi_i(p)]^2$, where M is the number of image components. The MSE is typically expressed in terms of the PSNR, which is defined as

$$\mathsf{PSNR} = 20 \log_{10} \left(\frac{2^\rho - 1}{\sqrt{\mathsf{MSE}}} \right), \tag{1}$$

where ρ is the number of bits/sample in each of the components of ϕ.

An important quantity used in our work is the ***maximum-magnitude second-order directional derivative (MMSODD)*** [3]. The MMSODD $\sigma(x, y)$ of a function f defined on $\mathbb{R}^2$ is defined as

$$\sigma(x, y) = \max\Big\{ |\alpha(x, y) + \beta(x, y)|, |\alpha(x, y) - \beta(x, y)| \Big\}, \tag{2}$$

where $\alpha(x, y) = \frac{1}{2}\left[\frac{\partial^2}{\partial x^2} f(x, y) + \frac{\partial^2}{\partial y^2} f(x, y)\right]$ and $\beta(x, y) = \sqrt{\frac{1}{4}\left[\frac{\partial^2}{\partial x^2} f(x, y) - \frac{\partial^2}{\partial y^2} f(x, y)\right]^2 + \left[\frac{\partial^2}{\partial x \partial y} f(x, y)\right]^2}$. In practice, the 1-dimensional (1-D) second-order derivative operators $\frac{\partial^2}{\partial x^2}$ and $\frac{\partial^2}{\partial y^2}$ can be computed using the filter with the transfer function $z - 2 + z^{-1}$. The derivative operator $\frac{\partial^2}{\partial x \partial y}$ is computed as the tensor product of two 1-D first-order derivative operators with the transfer function $\frac{1}{2}z - \frac{1}{2}z^{-1}$. In order to reduce the influence of noise, a third-order binomial lowpass filter is used to smooth each image component when computing any first- or second-order derivatives in our work.

III. Proposed Method

As mentioned earlier, a key component of our proposed method is Floyd-Steinberg error diffusion (FSED), a well-known technique originally proposed for digital halftoning. FSED selects points on a rectangular grid such that the selected points are distributed with a density that (approximately) matches that of a given function d, with the total number of selected points being controlled by a threshold τ.

In the context of our work, we use FSED to select a set P of sample points having the desired size N to within a tolerance ϵ (i.e., $||P| - N| \leq \epsilon$). The mapping of the threshold τ to the selected number of points is approximately monotonic. Thus, a binary search can be used to determine the value of τ that is needed to achieve a particular desired number N of selected points. Since it may not always be possible to find a value of τ that exactly satisfies $|P| = N$, we allow $|P|$ to differ from N by up to a small tolerance ϵ. In our work, we choose $\epsilon = 10$. More specifically, the binary search method finds the value to use for τ as follows. Find two choices τ_1, τ_2 of τ such that τ_1 results in $|P| > N$ and τ_2 results in $|P| < N$. Then, perform a binary search in $[\tau_1, \tau_2]$ to find a value of τ that results in $||P| - N| \leq \epsilon$.

FSED also requires a diffused-in error $\tilde{e}(i)$ for $i \in [0..W)$ to be specified as initial conditions for the algorithm. In halftoning, the initial diffused-in error is normally chosen as zero. In our application, choosing the initial diffused-in error as zero can lead to an undesirable startup effect that results in poor point selection near the start of the algorithm. Thus, we propose a new way to choose the initial diffused-in error to mitigate this startup effect. For a given a density function d for an image ϕ of width W and height H and a threshold

τ, we propose that the diffused-in error $\tilde{e}(i)$ for $i \in [0..W)$ be computed as follows:

1) Let $d_m(x, y) = d(x, H - 1 - y)$ for $(x, y) \in \Lambda$ (i.e., the function d_m is the function d flipped vertically);
2) Invoke FSED with density function d_m, threshold τ, and zero for the diffused-in error (where FSED starts from the 0th row of d_m). For the last (i.e., $(H - 1)$th) row processed by FSED, record the diffused-out error $\tilde{e}_o(i)$ for $i \in [0..W)$.
3) Let $\tilde{e}(i) = \tilde{e}_o(i)$ for $i \in [0..W)$.

With the above said, we can now introduce our proposed mesh-generation method, known as CMG. Although we only consider the generation of mesh models of RGB color images herein, it turns out that our CMG method can handle images with any arbitrary number M of components. So, in the interest of generality, we will initially present our method for arbitrary M, and then focus exclusively on the case that $M = 3$ (i.e., the RGB color case) in our subsequent analysis. Given an M-component image ϕ sampled on a truncated 2-D integer lattice Λ of width W and height H and a desired mesh size N, our method produces a mesh model of ϕ having a set P of sample points, where $|P| = N$, and the associated triangulation T. More specifically, our method consists of the following steps (in order):

1) Initial sample-point selection. Select a set P of $N_0 = \min\{\gamma N, WH\}$ sample points, where γ is a real constant satisfying $\gamma \geq 1$, by proceeding as follows (in order):
   a) For each $i \in [0..M)$, let σ_i be the MMSODD of the image component ϕ_i, where MMSODD is as defined earlier in (2).
   b) Compute the density function d, given by $d(x, y) = a(x, y)/(a_{\max} + \varepsilon)$, where $a(x, y) = \max\Big\{\sigma_0(x, y), \sigma_1(x, y), ..., \sigma_{M-1}(x, y)\Big\}$, $a_{\max} = \max_{(x,y)\in\Lambda} a(x, y)$, and ε is a small positive real constant (e.g., 10^{-12}).
   c) Invoke FSED with the binary search method and the new initial condition selection approach that we introduced previously, by taking d as the input density function and N as the desired number of sample points. Let P be the set of sample points output by FSED.
2) Initial mesh construction. Construct the Delaunay triangulation T of P, where T can be uniquely determined from P by using the preferred-directions technique [8].
3) Top of mesh-simplification loop. If $|P| \leq N$, output P and the associated triangulation T, and stop.
4) Choose a point p^* to delete from the mesh, as given by

$$p^* = \arg\min_{p \in P \setminus H} \Delta e(p),$$

   where H is the set of the (four) extreme convex hull points of Λ, $\Delta e(p)$ is defined as

$$\Delta e(p) = \sum_{i=0}^{M-1} \sum_{q \in \Lambda \cap R} [r^2_{i,P\setminus\{p\}}(q) - r^2_{i,P}(q)],$$

   R is the set of points belonging to the region of the triangulation affected by the deletion of p, and $r_{i,S}(p)$ denotes the approximation error at p for the component i for a mesh with the set S of sample points. Remove p^* from the triangulation T (i.e., let $P := P \setminus \{p^*\}$).
5) Go to step 3 (i.e., the beginning of the mesh-simplification loop).

Our above method has a single parameter γ that appears in step 1. Based on experimentation and other analysis, we advocate two possible choices for γ: 1) $\gamma = 1$; or 2) $\gamma = 4$. The choice $\gamma = 1$ yields lower computational/memory cost at the expense of lower mesh quality, whereas the choice $\gamma = 4$ yields greatly improved mesh quality at a modest increase in computational/memory cost. For convenience in the remainder of this paper, we use CMG(x) to denote our CMG method with the γ parameter chosen as x (e.g., CMG(1) denotes our CMG method with $\gamma = 1$).

IV. Results

Having introduced our proposed CMG method, we now evaluate its performance in terms of mesh quality and computational/memory cost by comparing it to the CED, CGPRFSED, and CGPR schemes introduced in Section I (i.e., the trivially color-extended versions of the ED, GPRFSED, and GPR methods). In order to offer the reader some further insights into our CMG method, we also present and comment on some experimental results relating to the choice of the γ parameter in this method. Lastly, we also include some results to show the effectiveness of our proposed approach for selecting the initial diffused-in error for FSED. The software implementations of the various methods used in this evaluation were developed by the authors of this paper and are written in C++.

A. Mesh Quality

In order to evaluate mesh quality, we employed a set of 45 RGB color images, taken mostly from the well-known JPEG-2000 [9], USC [10], and CIPR [11] test sets. For each of the 45 RGB color images in our test set and five sampling densities per image, we generated a mesh using each of the methods under consideration. In each test case, the difference between the reconstructed image obtained from the mesh and the original image was measured in terms of PSNR as defined in (1). The methods were ranked for each test case from best to worst, with a value of 1 corresponding to the best result. The average and standard deviation of the ranks were computed across each sampling density as well as overall with the results shown in Table I(a). A representative subset of the results for some of the individual test cases is shown in Table I(b). In what follows, we make various comparisons based on data in these tables.

CMG(1) vs. CED. Since our CMG(1) approach (i.e., our CMG method with $\gamma = 1$) can be viewed as an extension of the ED method to RGB color images with some additional improvements, we compare our CMG(1) approach to the CED scheme (i.e., the trivially color-extended version of the ED

TABLE I
COMPARISON OF THE MESH QUALITY OBTAINED WITH THE VARIOUS METHODS. (A) RANKINGS AVERAGED ACROSS 45 IMAGES. (B) APPROXIMATION ERRORS FOR INDIVIDUAL TEST CASES.

(a)

Samp. Density (%)	Average Rank*				
	CED	CMG(1)	CGPRFSED	CMG(4)	CGPR
0.5	4.88 (0.32)	4.12 (0.32)	2.83 (0.37)	**1.36** (0.48)	1.81 (0.70)
1.0	4.90 (0.29)	4.10 (0.29)	2.50 (0.50)	**1.12** (0.39)	2.38 (0.65)
2.0	4.90 (0.29)	4.10 (0.29)	2.24 (0.48)	**1.05** (0.21)	2.71 (0.50)
3.0	4.81 (0.39)	4.17 (0.43)	2.19 (0.45)	**1.05** (0.21)	2.79 (0.51)
4.0	4.81 (0.39)	4.19 (0.39)	2.14 (0.41)	**1.07** (0.26)	2.79 (0.51)
Overall	4.86 (0.34)	4.13 (0.35)	2.38 (0.51)	**1.13** (0.35)	2.50 (0.69)

*Standard deviations are given parentheses.

(b)

Image	Samp. Density (%)	PSNR (dB)				
		CED	CMG(1)	CGPRFSED	CMG(4)	CGPR
lena	0.5	17.48	19.18	25.63	26.04	**26.09**
	1.0	21.31	22.12	28.02	**28.38**	28.09
	2.0	25.54	25.77	30.33	**30.48**	30.13
	3.0	27.42	27.73	31.44	**31.68**	31.29
	4.0	28.82	28.83	32.13	**32.49**	32.04
pens	0.5	13.96	15.78	22.49	**24.05**	23.60
	1.0	17.24	19.27	25.95	**26.77**	26.40
	2.0	21.98	23.48	29.08	**29.43**	29.12
	3.0	25.05	25.91	30.59	**31.16**	30.77
	4.0	27.05	27.92	31.97	**32.45**	32.01
bluegirl	0.5	19.73	21.17	27.10	29.37	**29.68**
	1.0	22.49	25.30	31.99	**32.67**	32.54
	2.0	25.29	29.40	34.97	**35.38**	34.98
	3.0	29.49	31.90	36.39	**36.85**	36.33
	4.0	32.67	33.40	37.29	**37.86**	37.23

method introduced in Section I). Examining Table I(a), we can see that our CMG(1) approach clearly ranks better than the CED method at all sampling densities. More detailed analysis shows that our CMG(1) approach beats the CED method in 190/225 (84.4%) of the test cases by up to 7.08 dB. The individual results shown in Table I(b) are consistent with the overall ranking results, where our CMG(1) approach beats the CED method in 15/15 of the test cases by up to 4.11 dB. Consequently, in terms of mesh quality, our CMG(1) approach is clearly superior to the CED method.

CMG(4) vs. CGPRFSED. Since our CMG(4) approach (i.e., our CMG method with $\gamma = 4$) can be viewed as an extension of the GPRFSED method to RGB color images with some additional improvements, we compare our CMG(4) approach to the CGPRFSED scheme (i.e., the trivially color-extended version of the GPRFSED method introduced in Section I). Examining the overall results of Table I(a), we can see that our CMG(4) approach clearly ranks better than the CGPRFSED method at all sampling densities. More detailed analysis shows that our CMG(4) approach beats the CGPRFSED method in 220/225 (97.8%) of the test cases by up to 7.05 dB. The individual results shown in Table I(b) are consistent with the overall ranking results, where our CMG(4) approach beats the CGPRFSED method in 15/15 of the test cases by up to 2.88 dB. Consequently, our CMG(4) approach is superior to the CGPRFSED method in terms of mesh quality.

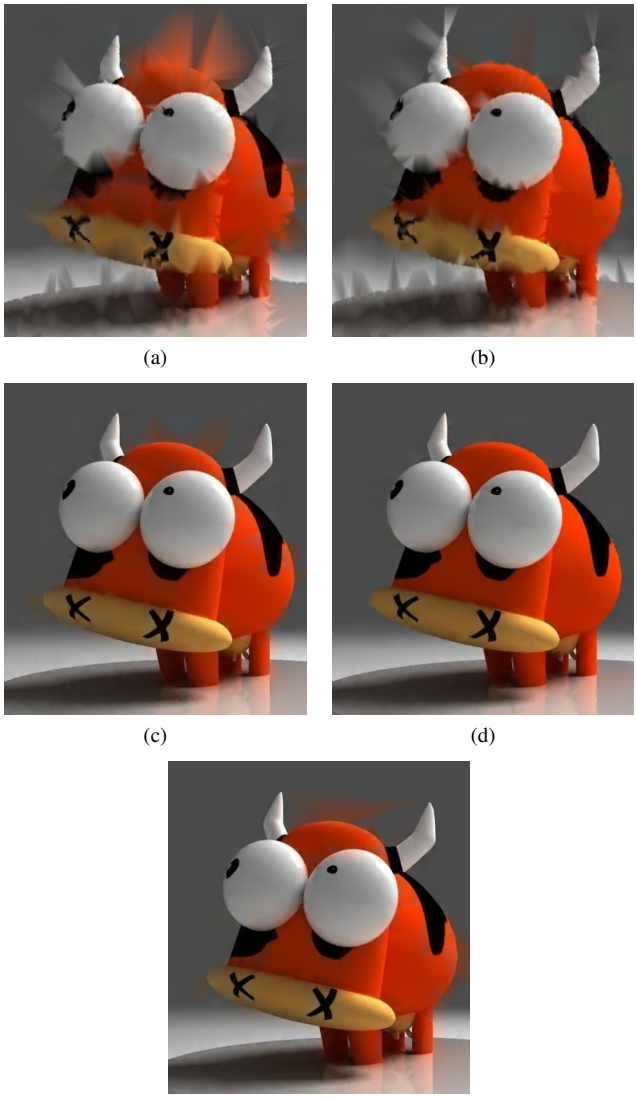
CMG(4) vs. CGPR. Since the CGPR method (i.e., the trivially color-extended version of the GPR method introduced in Section I) has an extremely high computational cost and our CMG(4) approach has higher computational cost than CMG(1), as will be seen later, we compare the performance of the CGPR method to our CMG(4) approach (in the interest of fairness). Examining Table I(a), we can see that, our CMG(4) approach has a better overall rank than the CGPR method, ranking clearly better than the CGPR method at all sampling densities. More detailed analysis shows that, our CMG(4) approach vastly outperforms the CGPR method, beating it in 200/225 (89.0%) of the test cases by up to 5.15 dB. Thus, our CMG(4) approach is vastly superior to the CGPR method in terms of mesh quality.

Subjective Quality. 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 considered. The set of reconstructed images obtained for one of the 225 test cases associated with Table I(a) is shown in Fig. 1. Examining Figs. 1(a) and (b), we can see that our CMG(1) approach (in Fig. 1(b)) yields a better image reconstruction than the CED method (in Fig. 1(a)). Examining Figs. 1(c), (d) and (e), we can see that the reconstructed image obtained from our CMG(4) approach (in Fig. 1(d)) is superior in quality compared to the images produced by the CGPRFSED and CGPR methods (in Figs. 1(d) and (e)), due to the difference in visual quality along the top boundary of the bull's head.

B. Remarks on the Choice of the γ Parameter

At this point, we would like to make a few comments concerning the choice of the γ parameter in our CMG method. As can be seen by the results of Table I, increasing γ from 1 to 4 significantly improves mesh quality. That is, CMG(4) typically yields meshes of much higher quality than CMG(1). Therefore, one might expect that choosing $\gamma = \infty$ (i.e., choosing the initial mesh to contain all WH sample points of the original image) would yield even better results than CMG(4). As it turns out, however, this is not the case. That is, choosing γ as large as possible does not typically yield the best mesh quality. In fact, experimentation has shown that the highest mesh quality is usually obtained when $\gamma \in [4, 5.5]$.

To better illustrate the above relationship between γ and mesh quality, we consider a representative example. In what follows, let D denote the sampling density of the (final) mesh to be generated and let D_0 denote the sampling density of the initial mesh (i.e., $D_0 = \frac{N_0}{WH}$) to be used during mesh generation. For one of our test images with a target sampling density $D = 2\%$, we measured the mesh quality (in terms of PSNR) as a function of the initial sampling density D_0, and present the result in Fig. 2. From Fig. 2, we can see



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

Fig. 1. Part of the reconstructed image obtained for the bull image at a sampling density of 0.5% with the (a) CED (25.23 dB), (b) CMG(1) (26.17 dB), (c) CGPRFSED (36.98 dB), (d) CMG(4) (39.61 dB), and (e) CGPR (36.53 dB) methods.

that as D_0 increases, the PSNR initially increases rapidly to a maximum value and then slowly decreases thereafter. In this particular case, the maximum value is achieved when $D_0 \approx 5D$ (i.e., for a γ value of 5). By considering many other test cases, we were able to determine that the best choice of γ typically lies in the range $[4, 5.5]$. In the interest of minimizing computational/memory cost, we elected to limit γ to 4 in our proposed method, leading us to propose CMG(4) (in addition to CMG(1)).

C. Computational and Memory Complexity

Earlier, we made several claims about the computational and memory costs of the various methods considered. Now, we present some results to substantiate those claims.

Computational Complexity. To begin, we consider the computational cost of the various methods under evaluation.

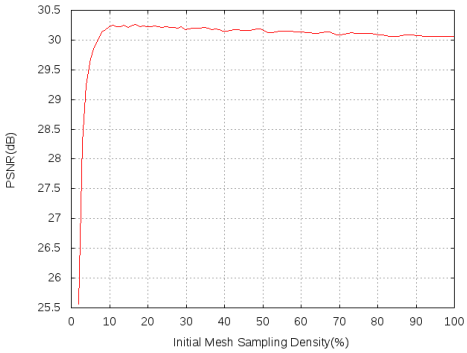


Fig. 2. Effect of varying the initial sampling density D_0 on mesh quality for the CMG method for the peppers image at a desired sampling density of 2%.

TABLE II
COMPARISON OF COMPUTATION TIME NEEDED BY THE VARIOUS METHODS FOR THE LENA IMAGE

Image	Samp. Density (%)	Time (s)				
		CED	CMG(1)	CGPRFSED	CMG(4)	CGPR
lena	0.5	0.16	0.35	0.68	1.72	29.97
	1.0	0.18	0.46	1.03	2.09	29.62
	2.0	0.23	0.58	2.31	2.79	29.33
	3.0	0.30	0.73	3.15	3.49	29.19
	4.0	0.38	0.80	4.32	5.04	29.00

For this purpose, we measured the execution times required for the various methods considered for the test cases in Table I, using a three-year-old laptop with a 1.60 GHz Intel Core i5 processor and 4 GB of RAM. A representative subset of these execution times is given in Table II.

Examining the results of Table II, we can see that the execution times required for our CMG(1) and CMG(4) approaches are quite similar to their main competitors CED and CGPRFSED, respectively. Moreover, for the results shown in Table II, our CMG(4) approach requires about 6 to 17 times less execution time than the CGPR method. In other words, although (as seen previously) our CMG(4) approach yields meshes with quality better than (or occasionally comparable to) the CGPR method, this is achieved at a significantly lower computational cost.

Memory Complexity. Next, we compare the memory complexities of the various methods considered. For each of the CED, CGPRFSED, CGPR, and CMG methods, memory usage is approximately proportional to the mesh size (i.e., the number of vertices in the mesh). Consequently, the peak memory usage of each method is approximately proportional to the peak mesh size. For an image of width W, height H, and sampling density D: the CMG(1) and CED methods each have a peak mesh size of DWH, which is the lowest of the methods considered; the CMG(4) and CGPRFSED methods each have a peak mesh size of $4DWH$, which is the next lowest; and finally, the CGPR method has a peak mesh size of WH, which is the highest of all of the methods. Thus, our CMG(1) and CMG(4) approaches requires similar memory costs to their main competitors CED and CGPRFSED, respectively. Clearly,

TABLE III

COMPARISON OF THE MESH QUALITY OBTAINED WITH THE DIFFERENT INITIAL-CONDITION SELECTION APPROACHES FOR FSED. (A) FRACTION OF CASES IN WHICH THE PROPOSED APPROACH BEATS THE CLASSICAL APPROACH (I.E., WIN RATIO) ACROSS 45 IMAGES. (B) APPROXIMATION ERRORS FOR INDIVIDUAL TEST CASES.

(a)

Samp. Density (%)	Win Ratio (%)	
	CMG(1)	CMG(4)
0.5	77.8	82.2
1.0	80.0	68.0
2.0	68.9	53.3
3.0	64.4	48.9
4.0	66.7	44.4
Overall	71.6	59.6

(b)

Image	Samp. Density (%)	PSNR(dB)			
		CMG(1)		CMG(4)	
		Proposed	Classical	Proposed	Classical
lena	0.5	**19.18**	18.00	**26.04**	25.83
	1.0	**22.12**	21.66	**28.38**	28.23
	2.0	**25.77**	25.56	30.48	**30.50**
	3.0	**27.73**	27.39	31.68	**31.71**
	4.0	**28.83**	28.57	32.49	**32.49**
pens	0.5	**15.78**	15.02	**24.05**	23.37
	1.0	**19.27**	18.17	**26.77**	26.30
	2.0	**23.48**	22.68	29.43	**29.45**
	3.0	**25.91**	25.80	**31.16**	31.15
	4.0	**27.92**	27.20	**32.45**	32.44
bluegirl	0.5	**21.17**	19.49	**29.37**	27.08
	1.0	**25.30**	22.65	**32.68**	31.38
	2.0	**29.40**	25.72	**35.38**	35.28
	3.0	**31.90**	29.08	**36.85**	36.84
	4.0	**33.40**	31.82	**37.86**	37.85

our CMG(4) approach requires much less memory than the CGPR method. For example, for D in the range 0.5% to 4.0%, our CMG(4) approach requires 6 to 50 times less memory than the CGPR method. Thus, our CMG method compares very favorably to other methods in terms of memory cost.

D. Evaluation of Proposed Initial-Condition Selection Strategy for FSED

As mentioned earlier, our CMG method employs a new strategy for selecting the initial diffused-in error for FSED. We now show this new strategy to be more effective than the classical approach of setting the diffused-in error to zero. In the case of each of the CMG(1) and CMG(4) methods, for each of the 45 images with five sampling densities per image, we generated a mesh using each of the proposed and classical strategies for selecting the diffused-in error for FSED, and we measured the mesh quality in terms of PSNR. In the case of each of the CMG(1) and CMG(4) methods, for each sampling density as well as overall, we calculated the fraction of cases that our proposed strategy beats the classical approach. The overall results are shown in Table III(a). A representative subset of the results for individual test cases is shown in Table III(b).

Examining Table III(a), we can see that, in the case of both CMG(1) and CMG(4), our proposed initial-condition selection strategy for FSED beats the classical one in terms of overall performance. A more detailed analysis shows that, in the case of CMG(1), our proposed approach beats the classical one at all sampling densities in 161/255 (71.6%) of the test cases by up to 3.87 dB. Moreover, in the case of CMG(4), our proposed approach beats the classical approach at sampling densities of 1.0% and lower in 68/90 (75.6%) of the test cases by a margin of 0.02 to 2.29 dB, and behaves similarly to the classical approach at higher sampling densities. Consequently, our proposed strategy for selecting the initial diffused-in error for FSED is superior to the classical one, especially at lower sampling densities.

V. CONCLUSIONS

In this paper, we proposed a new method for generating mesh models of RGB color images, known as CMG, which is based on ideas from the GPRFSED scheme. By varying the initial mesh size (through the γ parameter), our proposed method can easily tradeoff between mesh quality and computational/memory complexity. Based on experimentation and analysis, we recommended two particular choices of γ for selecting initial mesh size, where the higher choice of γ yields better mesh quality with a modest increase in computational/memory cost. Through experimentation, our CMG method was shown to outperform the CED, CGPRFSED, and CGPR methods in terms of mesh quality, while requiring lower or comparable computational and memory costs. In addition, our proposed method can also be directly used to generate mesh models for images with an arbitrary number of components. As part of our work, we also proposed a new initial-condition selection strategy for FSED, which can be used to mitigate undesirable startup effects in the algorithm. Our CMG method can benefit the growing number of applications that employ mesh models of images.

REFERENCES

[1] 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.

[2] S. A. Coleman, B. W. Scotney, and M. G. Herron, "Image feature detection on content-based meshes," *Proceedings of IEEE International Conference on Image Processing*, vol. 1, pp. 844–847, 2002.

[3] Y. Yang, M. N. Wernick, and J. G. Brankov, "A fast approach for accurate content-adaptive mesh generation," *IEEE Transactions on Image Processing*, vol. 12, no. 8, pp. 866–881, 2003.

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

[5] M. D. Adams, "A flexible content-adaptive mesh-generation strategy for image representation," *IEEE Transactions on Image Processing*, vol. 20, no. 9, pp. 2414–2427, 2011.

[6] R. W. Floyd and L. Steinberg, "An adaptive algorithm for spatial greyscale," *Proceedings of the Society for Information Display*, vol. 17, no. 2, pp. 75–77, 1976.

[7] "Recommendation: Studio encoding parameters of digital television for standard 4:3 and wide screen 16:9 aspect ratios," https://www.itu.int/rec/R-REC-BT.601-7-201103-I/en, March 2011.

[8] 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.

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

[10] "USC-SIPI image database," http://sipi.usc.edu/database, 2016.

[11] "CIPR still images, Canon," http://www.cipr.rpi.edu/resource/stills/canon.html, 2002.