

A Flexible Content-Adaptive Mesh-Generation Strategy for Image Representation

Michael D. Adams, *Senior Member, IEEE*

Abstract

Based on the greedy-point removal (GPR) scheme of Demaret and Iske, a simple yet highly-effective framework for constructing triangle-mesh representations of images, called GPRFS, is proposed. By using this framework and ideas from the error diffusion (ED) scheme (for mesh-generation) of Yang et al., a highly effective mesh-generation method, called GPRFS-ED, is derived and presented. Since the ED scheme plays a crucial role in our work, factors affecting the performance of this scheme are also studied in detail. Through experimental results, our GPRFS-ED method is shown to be capable of generating meshes of quality comparable to, and in many cases better than, the state-of-the-art GPR scheme, while requiring very substantially less computation and memory. Furthermore, with our GPRFS-ED method, one can easily tradeoff between mesh quality and computational/memory complexity. A reduced complexity version of the GPRFS-ED method (called GPRFS-MED) is also introduced to further demonstrate the computational/memory-complexity scalability of our GPRFS-ED method.

Index Terms

Image representations, triangle meshes, mesh generation, greedy point removal, error diffusion.

I. INTRODUCTION

In the last several years, image representations based on adaptive (i.e., irregular) sampling have been receiving an increasing amount of attention from the research community [1]–[8]. This is largely due to the fact that, in many situations, such representations have numerous advantages over those

Copyright (c) 2010 IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending a request to pubs-permissions@ieee.org.

Manuscript received September 12, 2010; revised March 6, 2011; accepted March 7, 2011. 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 Dr. Charles Creusere.

The author is with the Department of Electrical and Computer Engineering, University of Victoria, Victoria, BC, V8W 3P6, Canada (tel: 1-250-721-6025; fax: 1-250-721-6052; e-mail: mdadams@ece.uvic.ca).

based on regular (e.g., truncated-lattice) sampling, including greater compactness and the ability to facilitate methods that yield higher quality results or (in some cases) have lower overall complexity. Some applications that can benefit from image representations based on adaptive sampling include filtering [9], feature detection [10], restoration [7], tomographic reconstruction [5], computer vision [11], pattern recognition [12], and image/video compression [2], [13]–[18], to name but a few.

Amongst the many classes of image representations based on adaptive sampling (e.g., [1], [19]–[26]), triangle meshes have received considerable attention, with those utilizing Delaunay triangulations proving to be particularly effective (e.g., [3], [6], [27]). Since images are usually sampled on a truncated lattice, if a triangle mesh is to be used to represent an image, a means is needed for choosing a good subset of the original sample points from which to form a mesh approximation. This is the so-called mesh-generation problem. Due to the necessity for solutions to this problem, mesh-generation methods are of fundamental importance.

Two highly effective mesh-generation methods proposed to date are the **greedy point-removal (GPR)** scheme of Demaret and Iske [4] (called “adaptive thinning” in [4]) and the **error-diffusion (ED)** scheme of Yang et al. [28]. The GPR method is state of the art in terms of its ability to produce very high quality meshes, but has extremely high computational and memory requirements. For example, in the recent paper [1, Figs. 4 and 5], the GPR scheme (called “adaptive thinning” therein) was shown to yield meshes of vastly superior quality in comparison to all of the other methods considered. On the other hand, the ED method produces much lower quality meshes than the GPR scheme, but is still generally considered to be quite effective due to its extremely low computational and memory complexities. In this paper, we introduce a flexible new mesh-generation framework called **GPR from subset (GPRFS)**, based on the GPR scheme. Then, using this framework, we propose two mesh-generation methods known as **GPRFS with ED (GPRFS-ED)** and **GPRFS via modified ED (GPRFS-MED)**. Both of these methods exploit ideas from the ED scheme in order to achieve much lower computational and memory complexities than the GPR scheme, but make different tradeoffs between mesh quality and computational/memory complexity. The GPRFS-ED method produces meshes of quality comparable to (or better than) the GPR scheme, at a greatly reduced cost in terms of computational and memory complexities. The GPRFS-MED method, which is effectively a reduced complexity version of the GPRFS-ED scheme and also can be viewed as a modified version of the ED scheme, makes further substantial reductions in complexity at the cost of lower mesh quality. In passing, we note that our work described herein has been partially presented in our conference paper [29].

The remainder of this paper is organized as follows. To begin, Section II provides some background

information on triangle meshes for image representation. Then, the focus shifts to mesh-generation schemes. In Section III, the ED method is introduced and studied in detail, with an emphasis on factors affecting the method’s performance, and in Section IV, the GPR method is presented along with its shortcomings. In Section V, our newly proposed mesh-generation framework, GPRFS, and mesh-generation methods, GPRFS-ED and GPRFS-MED, are presented. Section VI evaluates our GPRFS-ED and GPRFS-MED methods in relation to other schemes. Through experimental results, our methods are shown to produce excellent quality meshes when computational/memory cost is considered. Finally, Section VII concludes the paper with a summary of our key results and some closing remarks.

Before proceeding further, a brief comment is appropriate regarding some of the notation employed in this paper. The set of integers is denoted \mathbb{Z} . The cardinality of a set S is denoted $|S|$. Lastly, when presenting algorithms, the symbol “:=” is used to denote variable assignment.

II. TRIANGLE-MESH-BASED IMAGE REPRESENTATIONS

Consider an image f defined on the domain $I = [0, W - 1] \times [0, H - 1]$ that has been sampled at the points in the set $\Lambda = I \cap \mathbb{Z}^2$ (i.e., f has been sampled on a truncated two-dimensional integer lattice of width W and height H). Conceptually, the process of generating a triangle-mesh approximation of f consists of the following steps:

- 1) Select a subset $S = \{(x_i, y_i)\}$ of the set Λ of the original sample points, where S must contain the (four) extreme convex hull points of I (i.e., the four corners of the image bounding box) so that the triangulation of S covers the entire image domain I .
- 2) Construct a triangulation of S .
- 3) For each face of the triangulation, form an interpolant over the face.
- 4) Combine the interpolants for all of the faces from step 3 to produce a single interpolant \hat{f} defined over the entire image domain I .

In the context of our work, in step 3, a planar interpolant is employed. That is, given a face of the triangulation with vertices (x_i, y_i) , (x_j, y_j) , and (x_k, y_k) , and the corresponding sample values $z_i = f(x_i, y_i)$, $z_j = f(x_j, y_j)$, and $z_k = f(x_k, y_k)$, we form the unique planar interpolant passing through the points (x_i, y_i, z_i) , (x_j, y_j, z_j) , and (x_k, y_k, z_k) . In step 2, the Delaunay triangulation [30] is employed, due largely to its good properties for approximation. In many applications, it is desirable that the triangulation of S be uniquely determined by S alone, as this avoids the need for additional side information during the triangulation process. Unfortunately, the Delaunay triangulation is only guaranteed to be unique if no four points in S are cocircular, and this condition is unlikely to be satisfied in practice, since S

TABLE I
TEST IMAGES

Name	Size	Depth	Description
bull	1024 × 768	8	computer-generated bull
ct	512 × 512	12	tomography [34]
glasses	1024 × 768	8	raytraced glasses
lena	512 × 512	8	woman [33]
peppers	512 × 512	8	numerous peppers [33]
us	512 × 448	8	ultrasound [34]
wheel	512 × 512	8	antialiased color wheel

is a subset of the lattice \mathbb{Z}^2 . Therefore, to eliminate this potential nonuniqueness problem, we use the preferred-directions scheme of Dyken and Floater [31] to uniquely select a single triangulation from the set of all possible Delaunay triangulations of S .

In practice, the difficult part of the above mesh-generation process is step 1. In this step, for a given desired number N of sample points, we must choose a subset S of Λ such that $|S| = N$ and the resulting mesh approximation error is as small as possible—ideally, a global minimum. This is the part of the mesh-generation problem that we address in this paper. In our work, the squared error is used as the error metric and is expressed in terms of the peak-signal-to-noise ratio (PSNR), which is defined as $\text{PSNR} = 20 \log_{10}(M/d)$, where $d = [|\Lambda|^{-1} \sum_{i \in \Lambda} |\hat{f}(i) - f(i)|^2]^{1/2}$, $M = 2^\rho - 1$, and ρ is the sample precision in bits/sample. As a matter of terminology, we refer to the quantity $|S| / |\Lambda|$ as the **sampling density**. Since problems like the one above are known to be NP-hard [32], developing effective mesh-generation methods (i.e., methods that produce high-quality meshes at a reasonable computational/memory cost) is a challenging task.

Before proceeding further, a brief digression is in order regarding the (grayscale) test images used in our work. Although many images were employed for evaluation/testing purposes, the results presented herein focus primarily on the small representative subset of images listed in Table I, which consists of photographic, medical, and computer-generated imagery, originating from various sources, including the USC image database [33] and JPEG-2000 test set [34].

III. ERROR DIFFUSION (ED) METHOD

Before presenting our new approach, we first introduce two other related mesh-generation methods in detail. The first of these methods is the **error-diffusion (ED)** method proposed by Yang et al. [28]. For a given image f (which is defined on I and sampled at the points in Λ) and a desired number N of sample

points, the ED method uses Floyd-Steinberg error diffusion [35] to generate a set S of N sample points, distributed such that the local density of sample points at each point $(x, y) \in \Lambda$ is proportional to the largest magnitude second-order directional derivative of f at (x, y) . More specifically, the ED method consists of the following steps:

- 1) From f , compute the sample-point density function d defined on Λ given by $d(x, y) = \tilde{d}(x, y)/\tilde{d}_{\max}$, where $\tilde{d}_{\max} = \max_{(x,y) \in \Lambda} \tilde{d}(x, y)$ and $\tilde{d}(x, y)$ is the maximum magnitude second-order directional derivative of f at (x, y) .
- 2) Set the threshold τ to use for Floyd-Steinberg error diffusion to be $\tau_0 = \frac{1}{2N} \sum_{(x,y) \in \Lambda} d(x, y)$.
- 3) Convert d to a binary-valued function b using nonleaky Floyd-Steinberg error diffusion as described in [28] with the threshold τ .
- 4) Set S to the set of all points (x, y) for which $b(x, y) \neq 0$. Then, let $S := S \cup H$, where H is the set of the (four) extreme convex hull points of I .
- 5) If $|S|$ is close enough to N , stop; otherwise, adjust τ appropriately (i.e., if $|S| > N$, increase τ ; if $|S| < N$, decrease τ) and go to step 3.

In step 1, \tilde{d} is computed as given by [28, Corollary 1 and Equation (12)]

$$\tilde{d}(x, y) = \max\{|\alpha(x, y) + \beta(x, y)|, |\alpha(x, y) - \beta(x, y)|\}, \quad (1)$$

where $\alpha(x, y) = \frac{1}{2}[\frac{\partial^2}{\partial x^2} f(x, y) + \frac{\partial^2}{\partial y^2} f(x, y)]$ and $\beta(x, y) = \sqrt{\frac{1}{4}[\frac{\partial^2}{\partial x^2} f(x, y) - \frac{\partial^2}{\partial y^2} f(x, y)]^2 + [\frac{\partial^2}{\partial x \partial y} f(x, y)]^2}$.

The partial-derivative operators in the preceding equation are formed from the tensor product of one-dimensional derivative operators, where the discrete-time approximations of the one-dimensional first- and second-order derivative operators are computed using the filters with transfer functions $\frac{1}{2}z - \frac{1}{2}z^{-1}$ and $z - 2 + z^{-1}$, respectively.

It is important to note that the above method does not *explicitly* construct a mesh. That is, although a choice is made for S , the above method does not triangulate S to form an explicit mesh, nor is any interpolant explicitly generated. The method tacitly assumes that, when an explicit mesh is later constructed, a Delaunay triangulation will be used to triangulate S , and then, from this triangulation, an interpolant is formed using the approach described earlier in Section II. (Note that, although [28] considers both interpolating and non-interpolating mesh models, we only consider the interpolating case herein.)

Since the expression for \tilde{d} in (1) involves partial derivatives, which are to be determined by convolution, two questions naturally arise. First, since derivative operations are sensitive to noise, exactly how much benefit can be realized by including a smoothing operator in the convolution kernels used for derivative computation, and which smoothing operator might be the most effective for this purpose? Second, how

should the image boundaries be handled when computing convolutions? Unfortunately, in [28], Yang et al. did not clearly specify how to treat image boundaries during convolution, nor did they provide a detailed quantitative analysis of the impact of different choices of smoothing operator. For this reason, we consider both of these issues in more detail in what follows.

Although many choices of smoothing operators could be made, we elected to consider binomial filters [36] herein. The one-dimensional k th-order binomial (lowpass) filter (with zero-phase and unity DC gain) has the transfer function $H_k(z) = z^{(k-1)/2}(\frac{1}{2} + \frac{1}{2}z^{-1})^{k-1}$, where k is odd. Binomial filters are attractive, since they have reasonably-good frequency responses with very simple filter coefficients. For convenience, in what follows, we henceforth refer to smoothing using the (two-dimensional) smoothing operator obtained from the tensor product of two one-dimensional k th-order binomial filters as $B(k)$ smoothing. Note that, since $H_1(z) = 1$, $B(k)$ smoothing degenerates into the case of no smoothing when $k = 1$. Thus, for convenience, we often use the notation “ $B(1)$ ” as a synonym for no smoothing.

BOUNDARY-HANDLING STRATEGY. To begin, let us focus our attention on the choice of the boundary-handling strategy to be used during convolution. Since only separable filters are employed, we need only concern ourselves with boundary handling in one dimension. In our work, we considered three common boundary-handling strategies: 1) zero extension (i.e., the signal is padded with zeros); 2) constant extension (i.e., the first and last samples of the signal are repeated); and 3) symmetric extension (i.e., the signal is mirrored about its first and last sample points). For numerous images, sampling densities, and smoothing operators (including the degenerate case of no smoothing), a mesh was generated using the ED method in conjunction with each of the boundary-handling strategies under consideration. In each case, the mesh quality (in terms of PSNR) was measured. A representative subset of the results is shown in Table II, covering both the cases of smoothing and no smoothing, with the best result highlighted in each case. From these results, we observe that regardless of whether smoothing is employed, zero extension yields the highest quality meshes. In particular, zero extension outperforms constant and symmetric extension by about 0.6 to 2.3 dB in the no-smoothing case and 1.3 to 2.3 dB in the $B(3)$ smoothing case. More generally, considering our comprehensive set of results (and not just the subset presented here), we found zero extension to fairly consistently perform best, regardless of the image and sampling density.

A closer analysis shows that the relatively poor performance of constant and symmetric extension is due to their inability to place a sufficient number of sample points along the boundary of an image, which leads to very high distortion in the vicinity of image boundaries. This behavior is illustrated in Fig. 1. In this figure, for one of the test cases from Table II, we show the location of the sample points obtained using each of the boundary-handling strategies. Comparing the results for zero extension (in

TABLE II
COMPARISON OF BOUNDARY-HANDLING STRATEGIES IN THE ED METHOD FOR THE `peppers` IMAGE

Samp. Density (%)	PSNR (dB)					
	No Smoothing			$B(3)$ Smoothing		
	Zero Ext.	Const. Ext.	Sym. Ext.	Zero Ext.	Const. Ext.	Sym. Ext.
1	19.00	18.10	17.94	21.35	19.10	19.23
2	23.08	21.52	21.51	26.09	24.37	24.75
4	27.69	25.42	25.83	29.85	27.58	27.73
8	31.11	29.46	30.49	32.11	30.65	30.13

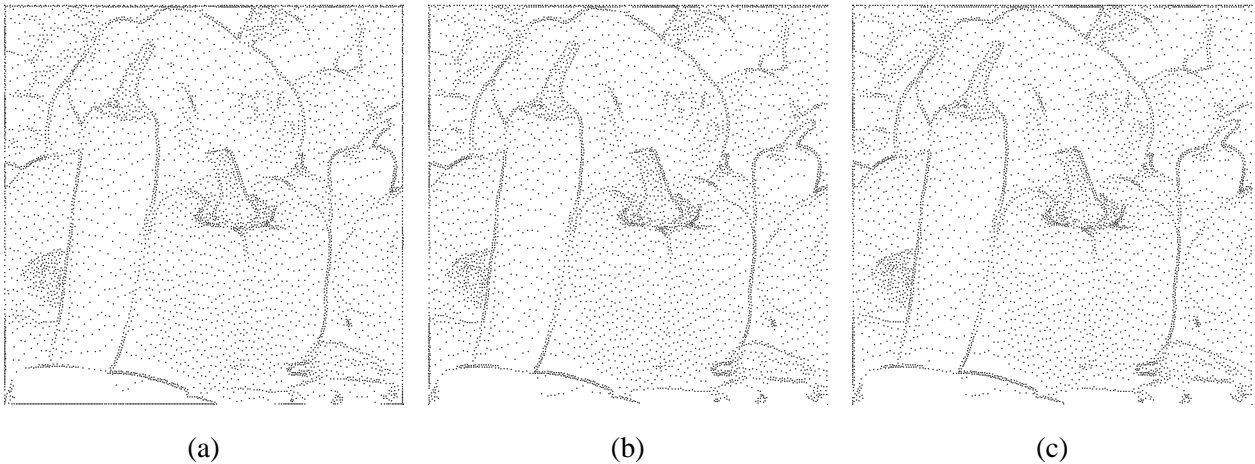


Fig. 1. Selected sample points obtained for the `peppers` image at a sampling density of 2% using $B(3)$ smoothing in conjunction with (a) zero extension, (b) constant extension, and (c) symmetric extension.

Fig. 1(a)) to those for constant and symmetric extension (in Figs. 1(b) and (c)), we observe that there is a significant difference in the number of sample points placed along the image boundary. In particular, in the case of constant and symmetric extension, noticeably fewer points are placed along the bottom and right boundaries of the image. Since zero extension clearly outperforms the other two boundary-handling strategies, zero extension is always used for the ED method in the remainder of this paper, unless otherwise noted.

SMOOTHING OPERATOR. Having examined various boundary-handling strategies, we now turn our attention to the choice of smoothing operator. In particular, we consider the use of $B(k)$ smoothing for $k \in \{1, 3, 5, 7, 9\}$ with $k = 1$ corresponding to the degenerate case of no smoothing. (Recall that the quantity k in $B(k)$ denotes the order of the smoothing filter.) For numerous images and sampling densities, a mesh was generated using the ED method in conjunction with each of the smoothing operators

under consideration (including the degenerate case of no smoothing). In each case, the mesh quality (in terms of PSNR) was measured. A representative subset of the results is shown in Table III, with the best result in each case being highlighted. From these results, we can make a number of observations. First, there is a general tendency for higher-order smoothing filters to perform better at lower sampling densities, while lower-order smoothing filters (such as $B(1)$ and $B(3)$) fare better at higher sampling densities. The preceding behavior is most clearly seen in the cases of the `bull`, `glasses`, `lena`, and `peppers` images. In the case of images having nonnegligible noise (e.g., photographic imagery), $B(k)$ smoothing with $k \geq 3$ generally performs much better than no smoothing, with $B(3)$ smoothing typically faring the best at sufficiently-high sampling densities. This behavior is clearly demonstrated by the `lena` and `peppers` images, where $B(3)$, $B(5)$, $B(7)$, and $B(9)$ smoothing consistently outperform no smoothing for all of the sampling densities considered. In instances where images have negligible noise and a significant amount of sharp edges, no smoothing often performs best, especially at higher sampling densities. This behavior is observed in the cases of the `ct` and `us` images. (As an aside, we note that although parts of the `ct` and `us` images are relatively smooth, these images still have a significant amount of sharp edges, due to factors such as an imaged patient region being superimposed on a larger rectangular background or the addition of text annotations, lines, and other markings.) In passing, we note that, in the above results, the subjective quality of images was generally found to correlate reasonably well with PSNR.

The above results show that, at sufficiently high sampling densities (which typically correspond to image reconstructions of high enough quality to be practically useful), either $B(3)$ or no smoothing tends to perform best. Since no smoothing performs very poorly in the case of images with noise, $B(3)$ smoothing arguably has the best overall performance across the broadest range of image types. For this reason, we recommend the use of $B(3)$ smoothing. Furthermore, in the remainder of this paper, we always employ $B(3)$ smoothing for the ED method, unless otherwise noted.

Having observed several trends in the above experimental results, we now seek to explain the reason behind these trends. As was observed above, in images with a nonnegligible amount of noise (such as photographic imagery like the `lena` and `peppers` images), not using smoothing tends to lead to poorer results. A more careful analysis shows that not employing smoothing in the presence of noise tends to lead to sample points with a somewhat more uniform (random) distribution, due to phantom large-magnitude derivatives resulting from noise. This behavior is illustrated in Fig. 2. In this figure, for one of our test cases, we show the location of the sample points chosen using no smoothing and $B(3)$ smoothing. Clearly, the points in Fig. 2(b), which were obtained with smoothing, better reflect the

TABLE III
COMPARISON OF SMOOTHING OPERATORS IN THE ED METHOD

Image	Samp. Density (%)	PSNR (dB)				
		None ($B(1)$)	Smoothing			
			$B(3)$	$B(5)$	$B(7)$	$B(9)$
bull	0.5	27.06	25.88	26.99	27.06	25.71
	1.0	31.30	33.33	32.53	32.55	31.79
	2.0	37.63	37.56	38.41	37.59	37.24
	3.0	39.97	40.36	39.65	39.83	39.58
ct	0.5	23.30	21.61	22.41	21.81	20.74
	1.0	31.96	29.45	25.60	25.57	25.18
	2.0	37.34	35.62	35.76	32.43	32.32
	3.0	40.13	39.40	38.30	37.23	34.11
glasses	1.0	19.59	21.30	21.85	22.45	22.81
	2.0	23.52	25.19	25.62	26.03	25.67
	3.0	26.01	27.59	27.80	27.41	27.56
	4.0	28.82	29.44	29.25	28.46	28.31
lena	1.0	20.12	21.13	22.08	22.87	23.21
	2.0	23.71	25.83	26.56	26.58	26.66
	3.0	26.23	28.05	28.36	28.14	28.01
	4.0	27.85	29.58	29.34	29.33	28.89
peppers	1.0	18.99	21.35	22.15	22.71	22.96
	2.0	23.07	26.08	26.24	26.53	26.47
	3.0	25.88	28.16	27.83	27.75	27.65
	4.0	27.68	29.84	29.19	28.47	28.49
us	3.0	19.29	17.62	21.11	20.49	20.39
	4.0	22.39	21.84	21.69	21.10	18.75
	5.0	24.17	22.32	22.30	21.99	22.00
	6.0	26.10	24.05	22.82	22.76	22.68

underlying structure of the image (especially fine detail such as image edges) than those in Fig. 2(a), which were obtained without smoothing.

To explain some of the other results from above, it is instructive to consider a simple example. For the wheel image shown in Fig. 3(a), we generated a mesh using $B(k)$ smoothing for $k \in \{1, 3, 9\}$, with the degenerate case of $k = 1$ corresponding to no smoothing. In each case, for the small rectangular region highlighted in Fig. 3(a), shown enlarged in Fig. 3(b), we present the resulting sample-point density function d (in Figs. 3(c) to (e)), image-domain triangulation (in Figs. 3(f) to (h)), and image approximation

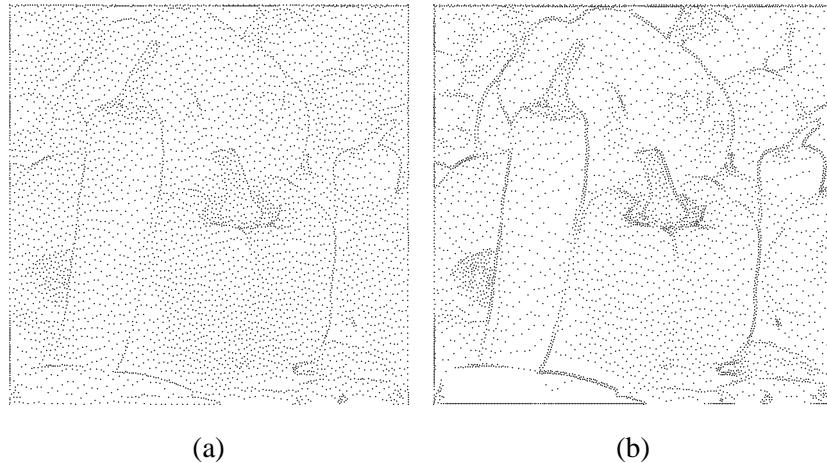


Fig. 2. Selected sample points obtained for the peppers image at a sampling density of 2% using (a) no smoothing and (b) $B(3)$ smoothing.

(in Figs. 3(i) to (k)). As is evident in Figs. 3(c) to (e), the sample-point density function generally has a double response to image edges. As one would expect with triangle-mesh approximations of images (and which is also demonstrated by Figs. 3(i) to (k)), the predominant type of artifact is the triangle tooth, which is caused by a triangulation edge crosscutting an image edge. When the crosscutting triangulation edge is short, a small tooth results (such as the numerous small teeth in Fig. 3(k)), and when the crosscutting edge is long, a large tooth is obtained (such as the one in Fig. 3(i)).

Examining the results of Fig. 3 more closely, we observe that, as the order of the smoothing filter increases: 1) The sample-point density function d becomes more blurred and the double response to each image edge widens, with the peaks in the response being displaced slightly farther away from the image edge. 2) In the image-domain triangulation, the sample points are spread more widely about image edges. 3) In the image reconstruction, the greater spread in the sample points about image edges leads to an increase in the number of smaller-teeth artifacts and a decrease in the number of larger-teeth artifacts. So, all other things being equal, higher-order smoothing filters tend to favor many smaller-teeth artifacts over a few larger-teeth artifacts (such as in Fig. 3(k)), whereas a lower-order smoothing filter tends to favor a few larger-teeth artifacts over many smaller-teeth artifacts (such as in Fig. 3(i)). Earlier, when examining the results of Table III, we observed a general tendency for higher-order smoothing filters to perform better at lower sampling densities, with lower-order smoothing filters becoming increasingly favored as the sampling density increases. We are now in a position to explain the reason for this behavior. At lower sampling densities, triangle edges tend to be longer and the mesh approximation error is most strongly influenced by large-teeth artifacts. Since a higher-order smoothing filter tends to trade

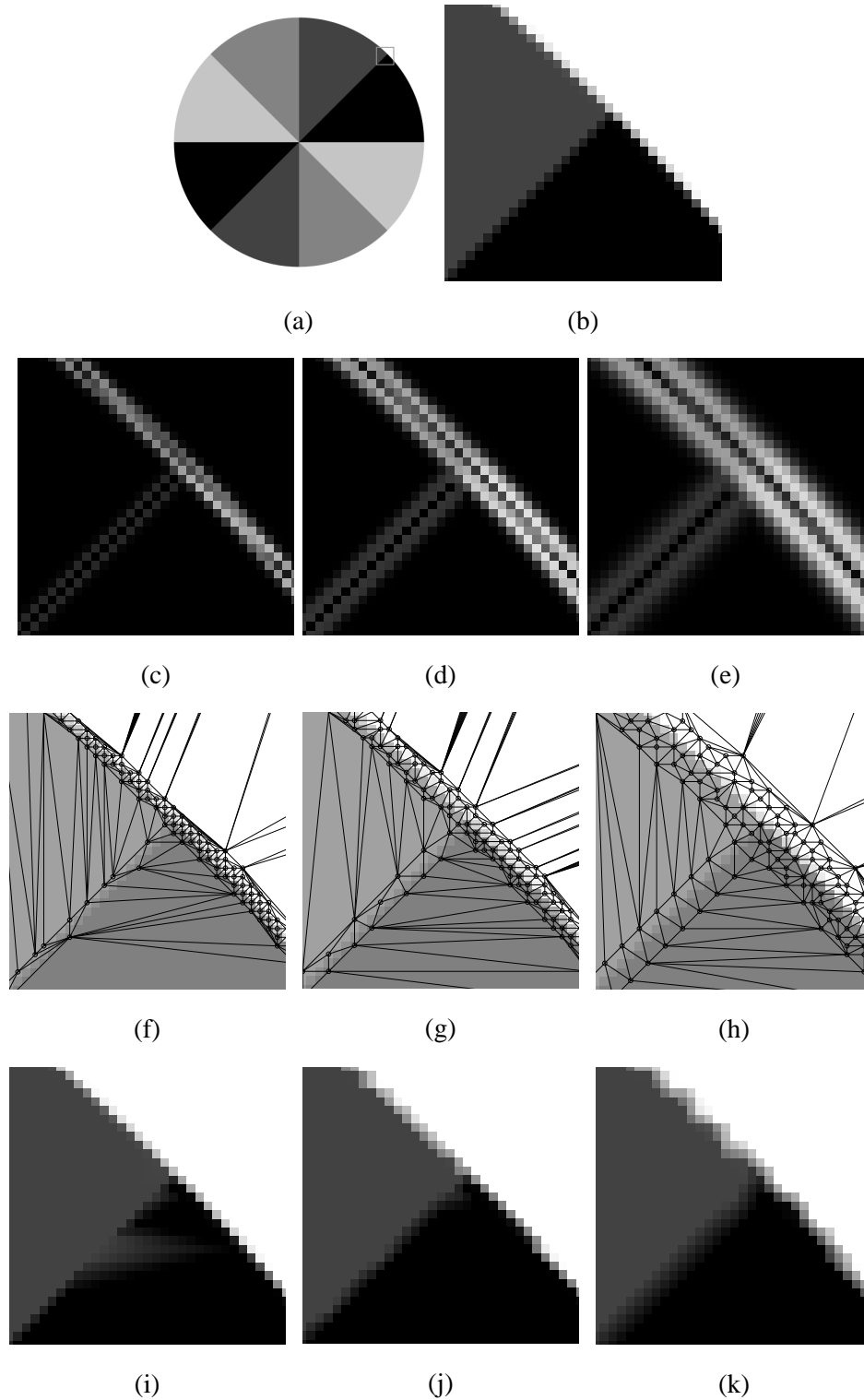


Fig. 3. The effect of smoothing on the generated mesh. (a) Full image, showing a rectangular region of interest. (b) Region of interest under magnification. The density function used for error diffusion with (c) $B(1)$ (i.e., no smoothing), (d) $B(3)$ smoothing, and (e) $B(9)$ smoothing; the corresponding image-domain triangulation for (f) $B(1)$, (g) $B(3)$, and (h) $B(9)$ smoothing; and the corresponding image approximation for (i) $B(1)$, (j) $B(3)$, and (k) $B(9)$ smoothing.

larger teeth for smaller ones, it has a mitigating effect on the error. Thus, higher-order smoothing filters tend to lead to better results at lower sampling densities. As the sampling density increases, however, large teeth are eliminated and small teeth become the main contributor to error. Consequently, at higher sampling densities, the approximation error is most strongly influenced by small teeth. Since higher-order smoothing filters encourage the creation of small teeth, such filters exacerbate the error incurred by small teeth, resulting in lower-order smoothing filters tending to perform better.

ERROR-DIFFUSION SCAN ORDER. In our analysis of the ED method, we also studied the effect that the scan order used for error diffusion has on mesh quality. More specifically, we considered the Hilbert scan order from [37] (where all error is diffused into the next point in the scan) as well as the raster and serpentine scan orders from [28]. For numerous test images and sampling densities, the ED method was employed to generate a mesh using each of the preceding three scan orders, and the resulting mesh quality measured (in terms of PSNR). A representative subset of the results is shown in Table IV. Generally, the serpentine scan order was found to fairly consistently outperform the raster scan order, regardless of the smoothing operator and particular image. So, in the remainder of this comparison, we focus our attention on the serpentine and Hilbert scan orders. The Hilbert scan order was sometimes found to perform better than the serpentine scan order at lower sampling densities. Although the Hilbert scan order can have an advantage at lower sampling densities, the serpentine scan order generally tends to be superior at higher sampling densities. This behavior is apparent in the results of Table IV. In particular, for each of the four images, the serpentine scan order consistently performs best at the highest sampling density. In practice, however, the sampling density is much more likely to be chosen near the higher end of the ranges considered in the table, since the lower sampling densities often correspond to relatively poor quality image reconstructions. Lastly, it was also observed that in some instances where the Hilbert scan order produced the best PSNR result, the corresponding subjective image quality was sometimes clearly not the best. Taking all of the above observations into consideration along with the fact that the serpentine scan order has much lower complexity than the Hilbert scan order (and essentially the same complexity as the raster scan order), we conclude that the serpentine scan order strikes the best overall balance between mesh quality and complexity. Consequently, the serpentine scan order is always used for the ED method in the remainder of this paper, unless otherwise noted.

For the reader's benefit, we note that the tendency of the Hilbert scan order to sometimes outperform the serpentine scan order at lower rates can be largely attributed to a startup effect that exists in error diffusion. In particular, if the sample-point density function d is quite small relative to the error-diffusion threshold τ in the region first processed by error diffusion, error will accumulate very slowly, resulting

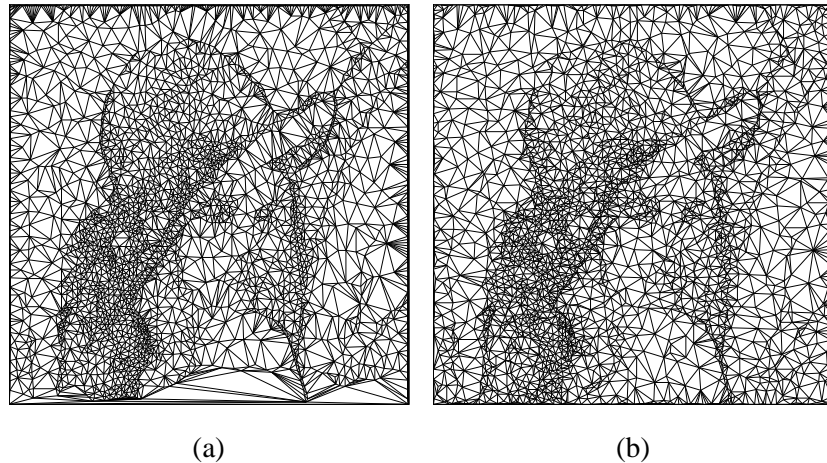


Fig. 4. Triangulations obtained for the `lena` image at a sampling density of 1% using the ED method with error diffusion employing the (a) serpentine and (b) Hilbert scan orders.

in very few (or no) sample points being selected in this region. Since the raster and serpentine scan orders process the image row-by-row starting from the bottom of the image, the entire bottom part of the image can potentially be affected by this startup behavior. For example, at low sampling densities, τ can become so large relative to d that it becomes either very unlikely or even impossible for any sample points to be placed in the first n rows processed during error diffusion, where n increases with τ . Since we process images starting from the bottom and proceeding upwards, this results in very few sample points in the bottom part of the image (except the two bottom corner points which are always forced to be selected). In contrast, the Hilbert scan order does not experience such a strong startup effect, due to the highly winding path that it employs. The above startup phenomenon is illustrated in Fig. 4. In this figure, for one of our test cases involving the `lena` image, we show the triangulations of the sample points chosen using each of the serpentine and Hilbert scan orders. Notice that, in the serpentine case (i.e., Fig. 4(a)), the number of sample points in the bottom part of the image (where the image intensity is changing more slowly) is abnormally small, and that no points were chosen on the bottom boundary of the image, except for the two bottom corner points which are always forced to be selected. This results in many sliver (i.e., long thin) triangles that lead to high mesh approximation error. On the other hand, in the Hilbert case (i.e., Fig. 4(b)), a much more reasonable distribution of sample points is obtained in the bottom part of the image.

SHORTCOMINGS. Although the ED method has very low computational complexity, it does have one major weakness. Namely, when selecting sample points, the ED method does not explicitly consider the error metric (i.e., squared error) or triangulation topology. To allow the reader to better appreciate the

TABLE IV
COMPARISON OF DIFFERENT SCAN ORDERS FOR THE ED METHOD

Image	Samp. Density (%)	PSNR (dB)		
		Raster	Serpentine	Hilbert
ct	0.5	19.00	21.61	24.28
	1.0	24.34	29.45	29.21
	2.0	35.59	35.62	29.95
	3.0	38.42	39.40	37.08
glasses	1.0	19.55	21.30	22.09
	2.0	24.07	25.19	25.70
	3.0	26.84	27.59	27.76
	4.0	28.74	29.44	29.22
lena	1.0	20.65	21.13	22.59
	2.0	24.95	25.83	25.77
	3.0	27.63	28.05	27.51
	4.0	29.16	29.58	28.96
peppers	1.0	20.03	21.35	22.96
	2.0	25.53	26.08	26.38
	3.0	27.94	28.16	28.41
	4.0	29.56	29.84	29.47

impact that this has on mesh quality, we make the following interesting observation, which is supported by experimental results presented later in Section V. Given a mesh produced by the ED method, we can typically remove, in an intelligent manner, a very substantial number of points from the mesh (in many cases, about half) without increasing the approximation error. In fact, by removing points, the approximation error can often be *reduced* significantly. At first, this result may seem surprising, but it does have a simple explanation. Because the ED method does not consider triangulation topology, a mesh produced by this method will typically have many triangle edges that crosscut image edges, resulting in higher approximation error. By removing points from the mesh, we can change the triangulation topology in such a way that triangle edges better align with image edges, resulting in lower approximation error.

IV. GREEDY POINT-REMOVAL (GPR) METHOD

The second mesh-generation method of interest herein is the **greedy point-removal (GPR)** scheme of Demaret and Iske [4] (called “adaptive thinning” in [4]), which is closely related to the adaptive thinning technique of Dyn et al. [8]. The GPR method first constructs a mesh that employs all of the sample

points of an image, and then repeatedly removes the sample point that yields the smallest increase in the squared error of the mesh approximation, until the desired number of sample points is obtained. More specifically, for an image sampled on a truncated two-dimensional integer lattice Λ of width W and height H and a desired number N of sample points, the method selects a set S of sample points as follows:

- 1) Initially, let $S := \Lambda$ (hence, $|S| = WH$).
- 2) Construct the Delaunay triangulation of S .
- 3) If $|S| \leq N$, output S and stop; otherwise, proceed to step 4.
- 4) For each point $p \in S$, compute the increase Δe_p in squared error of the mesh approximation that is incurred if p is removed from the triangulation. Note that, for the purposes of the preceding computation, it is assumed that a planar interpolant is formed over each face of the triangulation as described earlier in Section II.
- 5) For the point $p \in S$ that minimizes Δe_p , delete p from the triangulation, and let $S := S \setminus \{p\}$.
- 6) Go to step 3 (i.e., the beginning of the loop).

A few comments are now in order regarding the preceding algorithm. First, since the deletion of a vertex from a Delaunay triangulation is guaranteed only to affect the faces incident on the vertex to be deleted [38], step 4 can be performed quite efficiently in practice. That is, in each iteration *with the exception of the first*, step 4 only needs to recompute the error increase for a very small number of points, namely, the immediate neighbours of the point deleted in the previous iteration. Furthermore, by maintaining all of the triangulation vertices in a heap-based priority queue (where a vertex p has priority Δe_p), one can efficiently determine the vertex p minimizing Δe_p in step 5. For further details regarding efficient implementation, the reader is referred to [4], [8].

Although the GPR method has been shown to yield excellent quality meshes, it has one major weakness, namely its very high computational and memory costs. Since the GPR method starts with a triangulation containing all of the sample points of the image (i.e., $|\Lambda| = WH$ points), the mesh size at the beginning of the algorithm can be extremely large. For example, with today's digital cameras, a value for WH on the order of 10^7 is not unreasonable. This large initial mesh size leads to an algorithm that requires very high computation times and large amounts of memory.

Lastly, we note that, due to the greedy nature of the GPR method, it is extremely unlikely to yield a globally optimal solution. This suboptimality is a direct consequence of the short-sightedness of the greedy strategy. That is, when a point is removed, the algorithm fails to consider how this point's removal affects the evolution of the algorithm in *all subsequent* iterations (i.e., only the impact in the current iteration

is considered). In short, trying to minimize the increase in error in the current iteration may cause the error-increment values of later iterations to become much larger.

In light of the above suboptimality, it would seem plausible that solutions of quality comparable to (or perhaps even better than) those obtained with the GPR method could be achieved without the need to consider a mesh containing *all* WH sample points of the original image. That is, if we were to use a point-removal algorithm similar to that employed in the GPR method, but seed the algorithm with only a small subset of the WH sample points that are wisely chosen to be sufficiently close to a good optimal solution, such a scheme might be able to produce meshes of quality comparable to (or better than) those generated by the GPR method, but at only a small fraction of the computational and memory costs. It was precisely this hypothesis that led to the development of the new mesh-generation approach proposed in this paper, which we present next.

V. GREEDY POINT-REMOVAL FROM SUBSET (GPRFS) METHOD

With the above hypothesis as motivation, we propose a modified version of the GPR scheme, which we henceforth refer to as **GPR from subset (GPRFS)**. Given an image sampled on a truncated two-dimensional integer lattice Λ of width W and height H and a desired number N of sample points, our GPRFS method chooses a set S of sample points (where $|S| = N$) using an algorithm identical to the GPR scheme, except that step 1 (of the GPR scheme) is amended to read: “1) *Select a subset S_0 of Λ having size (i.e., cardinality) N_0 , where $N_0 \in [N, |\Lambda|]$ and the particular subset-selection policy to be employed for choosing S_0 , given N_0 , is specified as an input to the method. Then, let $S := S_0$. Note that S_0 must be chosen to include the (four) extreme points of the convex hull of I for the reasons mentioned earlier in Section II.*” In other words, our GPRFS method initially inserts an arbitrary subset of the sample points in Λ into the triangulation instead of always inserting all of them, as is the case in the GPR scheme. Note that our GPRFS method includes the GPR scheme as a special case. That is, if we choose $N_0 = |\Lambda|$, the GPR scheme is obtained. Since we can choose N_0 to have any value in the range $[N, |\Lambda|]$, our method has considerably more flexibility than the GPR scheme. For example, by choosing $N_0 \ll |\Lambda|$ (which results in a much smaller initial mesh), computational and memory costs can be greatly reduced relative to the GPR scheme. Also, as we will later show, by choosing $N_0 > N$ and $N_0 \ll |\Lambda|$, we can achieve meshes of quality comparable to, or better than, those produced by the GPR scheme. In this sense, we can tradeoff between mesh quality and computational/memory cost by varying N_0 in the range $[N, |\Lambda|]$.

CHOICE OF S_0 GIVEN N_0 . Of course, for our proposed approach to be useful, we need an effective subset-selection policy (i.e., a means for choosing S_0 , given N_0). Although many subset-selection policies

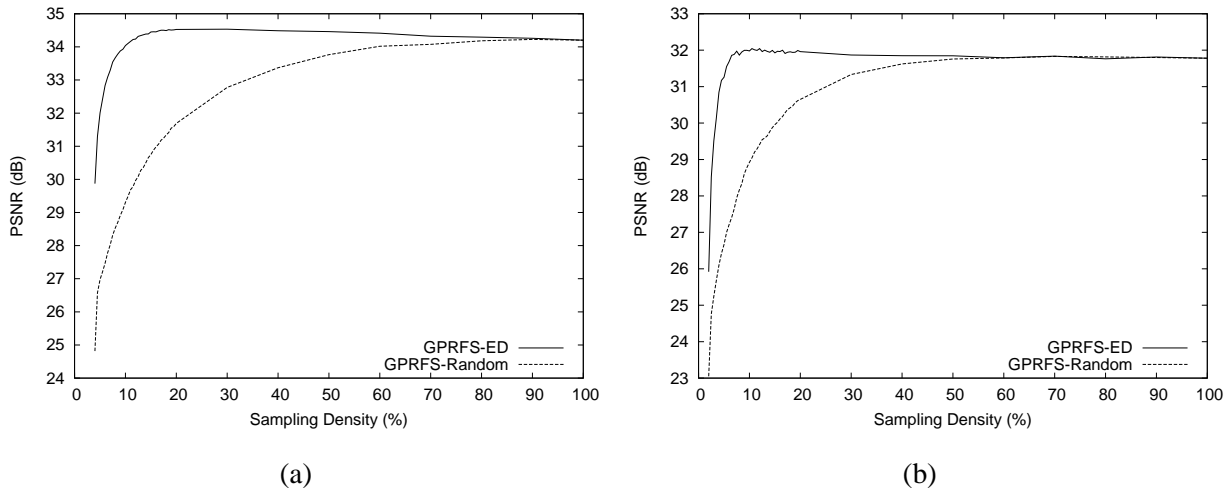


Fig. 5. Effect of varying the initial sampling density D_0 on mesh quality for the GPRFS-ED method. (a) The peppers image with a desired sampling density D of 4%; and (b) The lena image with a desired sampling density D of 2%.

could be developed, we consider only two herein, each leading to a different variant of the GPRFS method. The first policy, which is the one whose use is advocated by this paper, employs the ED method (of Section III) to select S_0 , and yields the GPRFS variant known as **GPRFS-ED**. In particular, the GPRFS-ED method employs the ED scheme with zero extension, $B(3)$ smoothing, and a serpentine scan order. The second policy, which is only used for benchmarking purposes, simply chooses S_0 randomly, and yields the GPRFS variant known as **GPRFS-Random**.

RECOMMENDED CHOICE OF N_0 . Although the GPRFS-ED and GPRFS-Random methods have been completely specified above, both methods have N_0 as an input parameter. Therefore, in the interest of making our work more practically useful, we provide a recommendation in what follows as to how one might reasonably choose N_0 , or equivalently, the initial sampling density D_0 (where N_0 and D_0 are related by $N_0 = D_0 |\Lambda|$). Since the rationale behind our recommendation is beneficial to know, a brief summary of the work that led to the recommendation is presented before the recommendation itself.

To help in determining an appropriate strategy for choosing D_0 , the following experiment was conducted for numerous images and values of the desired sampling density D (where $D = \frac{N}{WH}$). For the given image and value of D , we measured the mesh quality (in terms of PSNR) as a function of D_0 while keeping D fixed. Fig. 5 shows the results obtained for two test cases, with each graph having two lines, one for each of the two GPRFS variants. (For each graph, the horizontal axis corresponds to values of D_0 .)

From the results of Fig. 5, we can make a number of observations. The first is that the GPRFS-ED variant is vastly superior to the GPRFS-Random variant. That is, higher quality meshes (i.e., higher PSNR)

can be obtained with the GPRFS-ED variant than the GPRFS-Random variant for nearly all values of D_0 (and certainly all values of practical interest). In other words, using the ED scheme to wisely select S_0 in the GPRFS method is highly effective, much more so than choosing S_0 randomly. This shows the effectiveness of the subset selection strategy in our proposed GPRFS-ED method. Since the GPRFS-ED variant is vastly superior to the GPRFS-Random variant, we only consider the first of these variants in the remainder of our work. With this in mind, let us continue to examine Fig. 5, focusing our attention only on the results obtained with the GPRFS-ED variant. A careful examination of these results leads to the following interesting (and perhaps surprising) observation: In both graphs, the maximum PSNR is not obtained when D_0 is 100%, the point at which the GPRFS-ED method becomes equivalent to the GPR scheme. More specifically, as D_0 is decreased from 100% (corresponding to moving from right to left on the graphs), the PSNR climbs very slowly to a maximum value and then drops relatively rapidly thereafter. Due to this behavior, the GPRFS-ED method can be made to produce a higher quality mesh than the GPR scheme, provided that an appropriate choice of D_0 is made. Based on further experimentation, we found that, for D values of practical interest (e.g., $D < \frac{1}{10} = 10\%$), the GPRFS-ED method can usually be made to achieve a PSNR very close to that of the GPR method through a choice of D_0 in the approximate range $[4D, 5.5D]$. Consequently, we recommend that D_0 be chosen using a simple formula of the form $D_0 = \min\{\gamma D, 1\}$, or equivalently, that N_0 be chosen as

$$N_0 = \min\{\gamma N, WH\}, \quad (2)$$

where γ is a real constant satisfying $\gamma \geq 1$. Note that the GPRFS-ED method includes the ED scheme as a special case (i.e., the case of $\gamma = 1$). For photographic imagery (except at very low sampling densities), in order to achieve results comparable to the GPR scheme, it is typically sufficient to choose γ as 4, while in the remaining cases, a somewhat larger value (sometimes closer to 5.5) may be more appropriate. In the interest of minimizing computational/memory complexity, however, we recommend that γ be chosen nominally as $\gamma = 4$, incurring a small mesh-quality penalty relative to the GPR scheme in some cases. In the remainder of our paper, all experiments involving the GPRFS-ED method follow our recommendation of choosing N_0 using (2) with $\gamma = 4$, unless otherwise noted.

ADDITIONAL REMARKS ON GPRFS-ED. As has already been mentioned, the GPRFS-ED method uses the ED scheme with zero extension, $B(3)$ smoothing, and a serpentine scan order. This choice of options for the ED scheme, however, was made only after carefully studying the behavior of the ED scheme specifically when it is used *within* the GPRFS-ED method. In some respects, our findings were similar as to what was observed when the ED scheme was studied in isolation in Section III, but there were some differences. Since knowledge of these differences is likely to be beneficial to the reader, we

comment on these differences below.

Scan order. First, we comment on the choice of the scan order used for error diffusion. As in the case of the ED scheme (in isolation), in the GPRFS-ED method, the serpentine scan order was found to perform better than the raster scan order. In contrast to ED scheme, however, the percentage of cases in which the Hilbert scan order outperforms the serpentine and raster scan orders is greatly reduced to the point where any benefit of the Hilbert scan order is essentially eliminated. Recall that, in the context of the ED scheme, the good performance of the Hilbert scan order is largely due to its ability to reduce the degradation caused by startup effects in error diffusion. Because, for a given N (i.e., a desired number of sample points) the GPRFS-ED method employs a much smaller error-diffusion threshold τ than the ED scheme, the startup effect becomes much less of an issue in the GPRFS-ED method. Thus, the benefit of the Hilbert scan is essentially eliminated. For the above reasons, the serpentine scan order was ultimately chosen for use in our GPRFS-ED method.

Smoothing operator. Lastly, we comment on the choice of smoothing operator. To evaluate the performance of the various smoothing operators, we collected a set of results similar to those for the ED scheme appearing earlier in Table III. That is, for numerous images and sampling densities, a mesh was generated using the GPRFS-ED method in conjunction with each of the smoothing operators (including the degenerate case of no smoothing), and in each case, the mesh quality (in terms of PSNR) was measured. A representative subset of the results is shown in Table V. By comparing the results of Tables V and III, a few observations can be made. First, in the case of the GPRFS-ED method, $B(3)$ smoothing and no (i.e., $B(1)$) smoothing (especially the former) are much more strongly favored than in the ED case. This behavior can be briefly explained as follows. As was shown earlier in Section III, in the case of the ED method, one of the main advantages of higher-order smoothing filters is that they help to reduce the number of large triangle teeth artifacts, leading to lower mesh approximation error. Since the GPRFS-ED method effectively discards (via point-removal) bad sample points, such as those that produce large triangle teeth, the GPRFS-ED method does not benefit as much from the use of higher-order smoothing filters. For this reason, in the GPRFS-ED case, lower-order smoothing filters (namely, $B(1)$ and $B(3)$) are much more likely to perform best. Furthermore, by again comparing the results of Tables V and III, we can also observe that, in going from the ED method to GPRFS-ED method, the performance of $B(3)$ smoothing has improved overall relative to $B(1)$ smoothing. That is, in some cases where $B(1)$ smoothing was superior for the ED method, $B(3)$ smoothing is now best for the GPRFS-ED method; and in cases where $B(1)$ smoothing still beats $B(3)$ smoothing in the GPRFS-ED method, the margin (by which $B(1)$ smoothing outperforms $B(3)$ smoothing) tends to be smaller than with the ED method.

TABLE V
COMPARISON OF SMOOTHING OPERATORS IN THE GPRFS-ED METHOD

Image	Samp. Density (%)	PSNR (dB)				
		Smoothing				
		None ($B(1)$)	$B(3)$	$B(5)$	$B(7)$	$B(9)$
bull	0.5	40.14	39.78	40.63	39.99	39.19
	1.0	43.51	43.50	43.26	42.85	42.27
	2.0	45.70	45.63	45.50	45.29	44.94
	3.0	47.01	46.91	46.90	46.86	46.74
ct	0.5	37.17	36.85	35.76	33.49	33.32
	1.0	41.20	40.72	40.56	40.07	34.80
	2.0	45.10	44.63	44.19	44.13	43.99
	3.0	47.68	47.53	47.12	46.28	45.95
glasses	1.0	28.45	28.80	28.68	28.51	28.67
	2.0	32.32	32.44	32.07	31.73	31.55
	3.0	34.78	34.91	34.43	33.92	33.53
	4.0	36.74	36.90	36.41	35.79	35.37
lena	1.0	28.59	29.06	28.93	28.87	28.64
	2.0	31.65	31.94	31.83	31.68	31.50
	3.0	33.32	33.48	33.38	33.25	33.05
	4.0	34.44	34.56	34.45	34.29	34.11
peppers	1.0	29.03	30.00	29.76	28.99	28.98
	2.0	32.13	32.56	32.46	32.18	32.04
	3.0	33.41	33.73	33.65	33.53	33.39
	4.0	34.23	34.45	34.39	34.28	34.15
us	3.0	30.18	30.07	29.86	28.73	27.93
	4.0	32.61	32.57	32.70	32.33	31.37
	5.0	34.23	34.38	34.42	34.23	33.70
	6.0	35.68	35.75	35.66	35.52	35.17

Since $B(3)$ smoothing was found to yield the best results in the majority of cases, and is much better than $B(1)$ smoothing for images with noise (e.g., lena and peppers), $B(3)$ smoothing was ultimately chosen for use in our GPRFS-ED method.

GPRFS-MED. In the interest of providing an alternative tradeoff between mesh quality and complexity, we now present a slight variation on our GPRFS-ED method known by the name **GPRFS via modified ED (GPRFS-MED)**. The GPRFS-MED method is identical to the GPRFS-ED scheme, except for two key differences. First, instead of allowing flexibility in the choice of γ (in (2)), we always choose

$\gamma = 1$. Second, the stopping criterion in step 3 (of the GPRFS/GPRFS-ED method) is amended to read: “3) If $|S| \leq N$ and removing another point would cause an increase in the mesh approximation error, output S and stop; otherwise, proceed to step 4.” Essentially, our GPRFS-MED method is equivalent to adding a postprocessing step to the GPRFS-ED scheme with $\gamma = 1$ (or equivalently, the ED scheme) that continues to remove points as long as doing so does not cause the mesh approximation error to increase. This is precisely the intelligent point-removal process that was alluded to earlier in Section III (under the “Shortcomings” heading). Since the GPRFS-MED scheme chooses $\gamma = 1$, the initial mesh size is quite small and the computational cost is quite low relative to the GPRFS-ED method. The effectiveness of the GPRFS-MED method is a consequence of the typical behavior of mesh quality (in terms of PSNR) as points are removed from the mesh. This behavior is illustrated in Fig. 6. In particular, for two test cases, we show how the mesh approximation error changes as points are removed from the mesh (corresponding to moving from right to left on the graphs). The important observation to make from the graphs in the figure is that, as points are removed, the mesh quality (i.e., PSNR) increases significantly before starting to decrease. In fact, a very large fraction of the points from the initial mesh can be removed before the mesh quality first decreases (approximately 37% and 33% in the cases of Figs. 6(a) and (b), respectively). Furthermore, the point-removal process typically yields an overall *increase* in mesh quality. The reason for the preceding behavior is as was explained earlier in Section III. It is also important to note that the increase in PSNR resulting from point removal also correlates well with subjective image quality. This can be seen, for example, in the subjective results that appear later in Figs. 7(a) and (b) which correspond to the `peppers` test case from Fig. 6(a) above. In particular, Figs. 7(a) and (b) show, respectively, part of the reconstructed images obtained before and after application of the point-removal process. Notice how the triangle teeth artifacts are reduced in the reconstruction obtained after point removal has been applied. Lastly, although in principle we could also change the GPRFS-ED method so that its step 3 is amended as above, such a change would not have any significant effect on the GPRFS-ED method in practice. This is because, with this method, γ is typically chosen sufficiently large (i.e., $\gamma \geq 4$) that the mesh quality is essentially guaranteed to be decreasing by the time the current number of sample points is reduced to N (so the number of sample points would not be reduced beyond N).

VI. EVALUATION OF THE PROPOSED METHODS

Having introduced our GPRFS-ED and GPRFS-MED methods, we now compare their performance to the GPR and ED schemes in terms of mesh quality as well as computational and memory complexities. In passing, we note that the implementations of the ED, GPR, GPRFS-ED, and GPRFS-MED methods,

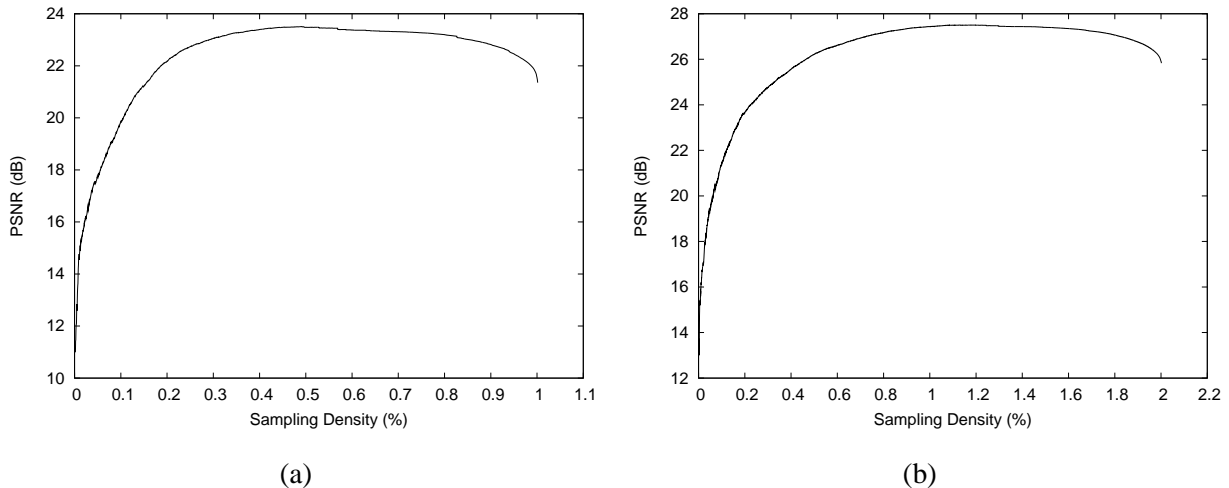


Fig. 6. Effect of removing points from the initial mesh in the cases of the (a) `peppers` image with an initial sampling density of 1%, and (b) `lena` image with an initial sampling density of 2%.

used for comparison purposes herein, were developed independently by the author in C++. The ED and GPR implementations follow the details provided in [28] and [4], [8], respectively. The GPRFS-ED and GPRFS-MED implementations were derived by extending the GPR implementation in a straightforward manner to handle different initial conditions (i.e., a different initial mesh). Lastly, for the benefit of future works that may wish to compare with results herein, the number N of sample points in a mesh can be computed from the sampling density D , and the width W and height H of the original image using the formula $N = \lfloor DWH \rfloor$.

MESH QUALITY. For several combinations of images and desired sampling densities, the GPRFS-ED, GPRFS-MED, GPR, and ED methods were used to generate meshes, and the mesh quality (in terms of PSNR) was measured. A representative subset of the results is given in Table VI, with the best result in each case being highlighted. For the GPRFS-ED method, two sets of results are provided, corresponding to two choices of the parameter γ (i.e., the nominal choice of $\gamma = 4$ as well as $\gamma = 5.5$). Generally, the PSNR was found to correlate reasonably well with subjective image quality. For the benefit of the reader, however, an example illustrating the subjective quality achieved by the various methods is provided in Fig. 7, where a small part of each mesh approximation is shown under magnification. In what follows, we examine the above results from Table VI and Fig. 7 in detail.

GPRFS-ED versus GPR and ED. To begin, we compare our GPRFS-ED method to the GPR and ED schemes. From Table VI, we can see that for photographic images (e.g., `lena` and `peppers`), our GPRFS-ED scheme (with either $\gamma = 4$ or $\gamma = 5.5$) yields results comparable to, or better than, the GPR scheme. In particular, for $\gamma = 5.5$, our GPRFS-ED method performs better at all sampling densities of

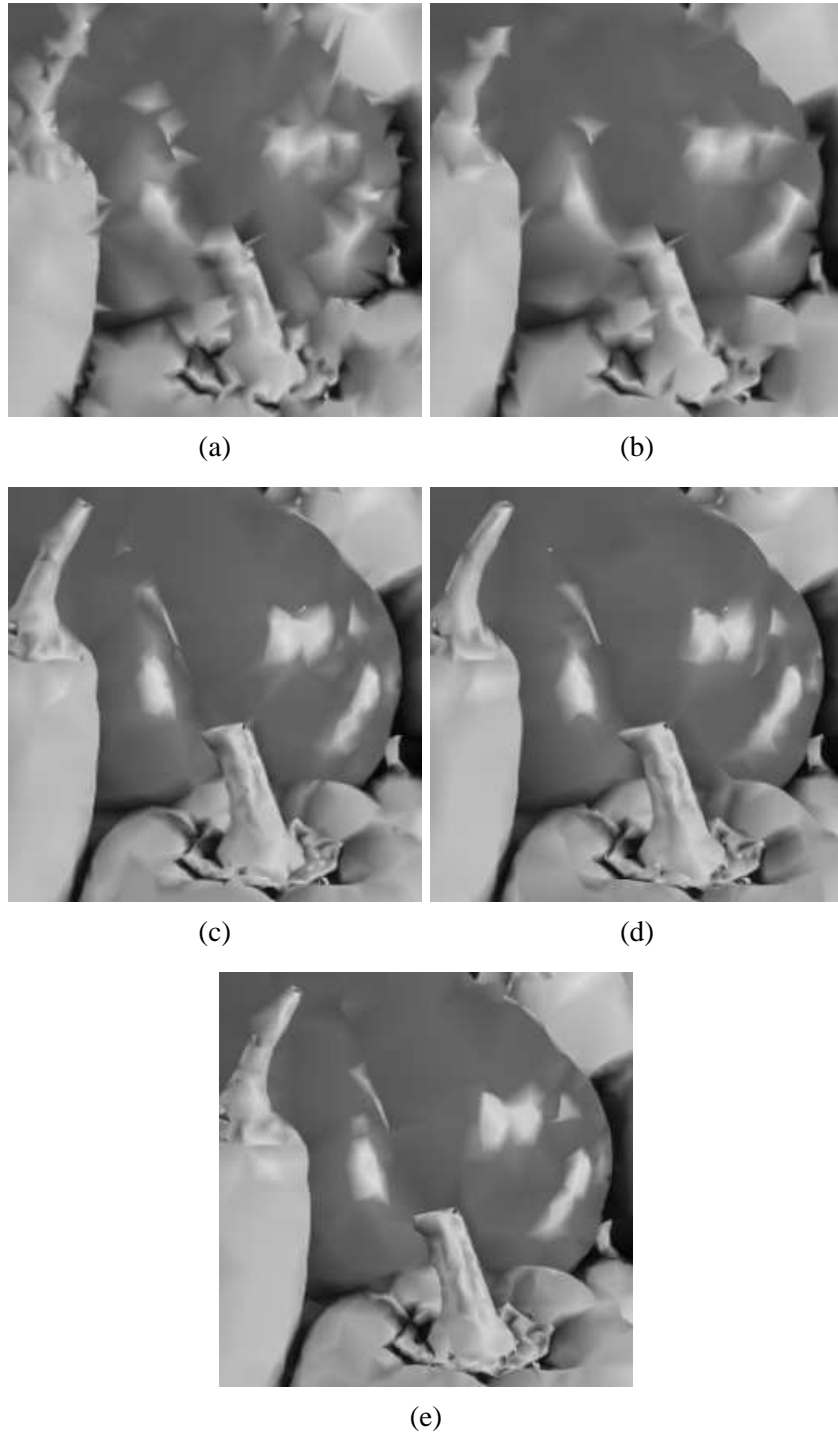


Fig. 7. Part of the image approximations obtained for the `peppers` image at a sampling density of 1% with the (a) ED (21.35 dB) (b) GPRFS-MED (23.34 dB, actual sampling density 0.64%) (c) GPRFS-ED with $\lambda = 4$ (30.00 dB), and (d) GPRFS-ED with $\lambda = 5.5$ (30.20 dB), and (e) GPR (30.04 dB) methods.

TABLE VI
COMPARISON OF MESH QUALITY OBTAINED WITH THE VARIOUS METHODS

Image	Samp. Density (%)	PSNR (dB)				
		ED	GPRFS- MED*	GPRFS-ED		GPR
				$\gamma = 4$	$\gamma = 5.5$	
bull	1.0	33.30	35.94 (0.54)	43.50	43.88	43.99
	2.0	37.56	39.63 (1.08)	45.63	45.84	45.80
	3.0	40.36	42.51 (1.69)	46.91	47.12	47.11
	4.0	41.51	43.74 (2.27)	47.97	48.16	48.22
lena	1.0	21.13	23.37 (0.63)	29.06	29.21	29.10
	2.0	25.83	27.43 (1.37)	31.94	32.00	31.77
	3.0	28.05	29.50 (2.07)	33.48	33.55	33.33
	4.0	29.58	30.87 (2.85)	34.56	34.60	34.39
peppers	1.0	21.35	23.34 (0.64)	30.00	30.20	30.04
	2.0	26.08	27.45 (1.34)	32.56	32.70	32.40
	3.0	28.16	29.62 (1.98)	33.73	33.82	33.50
	4.0	29.84	31.39 (2.65)	34.45	34.55	34.20
us	3.0	17.62	19.01 (2.10)	30.07	30.12	30.00
	4.0	21.84	24.00 (2.74)	32.57	32.68	32.48
	5.0	22.32	24.43 (3.43)	34.38	34.40	34.21
	6.0	24.05	26.06 (4.09)	35.75	35.78	35.65

*The numbers in parentheses are the actual sampling densities (in percent).

interest. For $\gamma = 4$, our GPRFS-ED method tends to be very slightly worse by a few hundredths of a dB at a sampling density of 1% (which is clearly still comparable) but better at higher sampling densities. As we will show later, however, at a sampling density of 1%, our GPRFS-ED method requires about 22 times less computation time and about 25 times less memory than the GPR scheme. So any small difference in mesh quality at a sampling density of 1% is arguably a small price to pay, considering the savings in computational/memory cost. For other types of imagery such as computer-generated (e.g., bull) and medical (e.g., us), our GPRFS-ED method still fares reasonably well relative to the GPR scheme, producing (in most cases) results close to the GPR scheme, especially when one considers the much lower complexity of our GPRFS-ED method. From the above results, we can also see that our GPRFS-ED method outperforms the ED scheme by a very large margin (i.e., by 4.6 to 12.5 dB in all cases), demonstrating that the excellent performance of our GPRFS-ED method is not simply due to its use of the ED scheme alone. From the reconstructed images shown in Figs. 7(c), (d), and (e), we can see that, in terms of subjective quality, our GPRFS-ED method is comparable to (if not better than) the

TABLE VII
SUMMARY RESULTS COMPARING THE GPRFS-ED METHOD TO THE GPR SCHEME FOR THE 24 PHOTOGRAPHIC IMAGES OF
KODAK TEST SET

Samp. Density (%)	GPRFS-ED $\gamma = 4$			GPRFS-ED $\gamma = 5.5$		
	Mean Diff.	Min. Diff.	Max. Diff.	Mean Diff.	Min Diff.	Max. Diff.
1.0	0.035	-0.191	0.223	0.169	-0.010	0.301
2.0	0.125	-0.037	0.266	0.176	0.056	0.287
3.0	0.119	-0.024	0.273	0.184	0.103	0.293
4.0	0.105	-0.033	0.228	0.162	0.062	0.234

GPR scheme.

To further demonstrate the effectiveness of our GPRFS-ED method for photographic imagery, we provide some statistical results taken over a larger set of images. For (grayscale versions of) each of the 24 images in the well-known Kodak test set [39], we generated meshes at several sampling densities using the GPRFS-ED and GPR methods and measured the mesh quality in terms of PSNR in dB. Then, we computed the difference between the PSNRs obtained for the GPRFS-ED and GPR methods in each case, with a positive value corresponding to better performance by the GPRFS-ED method. The resulting differences are summarized in statistical form in Table VII. Clearly, at all sampling densities, our GPRFS-ED method for both $\gamma = 4$ and $\gamma = 5.5$ outperforms the GPR scheme on average (i.e., the mean difference is positive, ranging from about 0.03 to 0.19 depending on the sampling density). The minimum and maximum differences show that, in terms of worst-case behavior, our GPRFS-ED method performs better than the GPR scheme as well (i.e., the magnitude of the minimum difference is always greater than the magnitude of the maximum difference). In fact, for $\gamma = 5.5$, our GPRFS-ED method almost beats the GPR scheme in every case (since the minimum difference is only very slightly less than zero). Thus, our GPRFS-ED method, in spite of its substantially lower complexity, produces meshes of quality comparable to, or better than, the GPR scheme.

GPRFS-MED versus ED. As will be seen shortly, the GPRFS-MED method has computational and memory complexities more comparable to the ED scheme than the GPR approach or GPRFS-ED method with $\gamma \in \{4, 5.5\}$. For this reason, it is most meaningful to compare the GPRFS-MED method to the ED scheme, as we do here. At this point, we revisit the results of Table VI from above. Since the GPRFS-MED method normally produces a mesh with a sampling density much less than the target (i.e., desired) sampling density, in order to allow a fair comparison, the actual sampling densities obtained with

the GPRFS-MED method are shown in parentheses in this table. From the results of the table, we can see that, for a given desired sampling density (corresponding to a single row in the table), our GPRFS-MED method typically produces a mesh with a PSNR about 1 to 2 dB higher than the ED scheme. Furthermore, this better performance is achieved in spite of the fact that the size of each mesh produced by the GPRFS-MED method is about 50% to 66% of the size of the corresponding mesh generated by the ED scheme. If we take this smaller mesh size into account, the performance of our GPRFS-MED method is even more impressive relative to the ED scheme. By comparing Figs. 7(a) and (b), it is clear that, in terms of subjective quality, our GPRFS-MED method is also vastly superior to the ED scheme (especially, when one considers the significantly lower sampling density in the GPRFS-MED case).

COMPUTATIONAL AND MEMORY COMPLEXITIES. Earlier, we made several claims about the computational and memory complexities of the various methods under consideration. We now present some results to substantiate these claims.

Computational complexity. First, we consider the computational complexities of the methods, where computational complexity is measured in terms of execution time. Since the GPR, GPRFS-ED, and GPRFS-MED schemes are all based on point removal, the times required for these methods are largely determined by the initial mesh size. So, we would expect methods with a larger initial mesh size to be slower. For several test images and desired sampling densities, we measured the time required to generate a mesh using each of the GPRFS-MED, GPRFS-ED, GPR, and ED methods. A representative subset of these results (for the `peppers` image) is shown in Table VIII. The most important comparison to be made here is between the GPRFS-ED and GPR methods. The above results show that, for $\gamma = 4$ and $\gamma = 5.5$ respectively, our GPRFS-ED method requires anywhere from about 9 to 33 and 6 to 25 times less computation time than the GPR scheme, with the difference being most pronounced at lower sampling densities. In other words, although (as seen previously) our GPRFS-ED method produces meshes of quality comparable to, or better than, the GPR scheme, this is accomplished with very substantially less computation. Comparing the time required by the GPRFS-MED and ED methods, we see that although GPRFS-MED scheme takes about 3 or 4 times longer than the ED scheme, the absolute time difference is less than one second. So, given the fact that the GPRFS-MED method yields meshes of vastly superior quality (typically, by a margin of about 1 or 2 dB) at significantly lower sampling densities relative to the ED scheme, the GPRFS-MED method is arguably quite competitive with the ED scheme. Lastly, we can observe that as γ is decreased, the complexity of our GPRFS-ED/GPRFS-MED method decreases. That is, as we progress from GPRFS-ED with $\gamma = 5.5$ to GPRFS-ED with $\gamma = 4$ to GPRFS-MED (i.e., $\gamma = 1$), we reduce computational complexity at the cost of lower mesh quality. (Recall, the GPRFS-MED

TABLE VIII
COMPARISON OF THE COMPUTATIONAL COMPLEXITIES OF THE VARIOUS METHODS FOR THE `peppers` IMAGE

Samp. Density (%)	Time (s)				
	ED	GPRFS- MED	GPRFS-ED		GPR
			$\gamma = 4$	$\gamma = 5.5$	
0.5	0.171	0.488	1.435	1.902	46.441
1.0	0.178	0.592	2.125	2.906	45.761
2.0	0.196	0.822	3.500	4.927	45.281
3.0	0.214	1.031	4.860	7.028	44.838

method is equivalent to the GPRFS-ED scheme with $\gamma = 1$ and a postprocessing step.) In this sense, our GPRFS-ED/GPRFS-MED method is highly scalable in terms of computational complexity (i.e., we can easily tradeoff between mesh quality and computational cost).

Memory complexity. Next, we compare the memory complexities of the various methods under consideration. In each of the GPRFS-ED, GPRFS-MED, and GPR methods, the memory usage is dominated by the mesh data structure and (to a lesser extent) a priority queue that contains one entry per mesh vertex. Furthermore, due to the similarities between these methods, they all employ identical data structures for representing the mesh and priority queue. Consequently, the peak memory usage for each method is approximately proportional to the peak number of mesh vertices. With the ED method (when used to actually construct a mesh), the peak memory usage is also dominated by the peak mesh size. Thus, for all of the methods, the peak mesh size is a good indicator of peak memory consumption. For sampling densities of practical interest, the peak mesh size for the various methods is as shown in Table IX, where W and H are the image width and height, and D is the desired sampling density. From the results of this table, we can see that for sampling densities from 1% to 4% the GPRFS-ED method with $\gamma = 5.5$ requires from $\frac{25}{5.5} \approx 4.5$ to $\frac{100}{5.5} \approx 18.1$ times less memory than the GPR scheme, with the difference being most pronounced at lower sampling densities. Similarly, the GPRFS-ED method with $\gamma = 4$ requires from $\frac{25}{4} \approx 6.2$ to $\frac{100}{4} \approx 25$ times less memory than the GPR scheme. Moreover, by using the GPRFS-MED variant, we can realize even greater memory savings (i.e., 25 to 100 times less memory). Clearly, our GPRFS-ED/GPRFS-MED method offers a very substantial memory savings relative to the GPR scheme. Again, as we move from GPRFS-ED with $\gamma = 5.5$ to GPRFS-ED with $\gamma = 4$ to GPRFS-MED (i.e., GPRFS-ED with $\gamma = 1$ and a postprocessing step), we tradeoff mesh quality for a reduction in memory requirements. In this sense, our GPRFS-ED/GPRFS-MED method is also highly scalable in terms of memory complexity.

TABLE IX
COMPARISON OF THE PEAK MESH SIZE FOR THE VARIOUS METHODS

Method	Peak Mesh Size	Relative Peak Mesh Size		
		General	$D = 1\%$	$D = 4\%$
ED	DWH	1	1	1
GPRFS-MED	DWH	1	1	1
GPRFS-ED, $\gamma = 4$	$4DWH$	4	4	4
GPRFS-ED, $\gamma = 5.5$	$5.5DWH$	5.5	5.5	5.5
GPR	WH	$1/D$	100	25

VII. CONCLUSIONS

In this paper, we have proposed a flexible new framework for content-adaptive mesh generation for image representation, called GPRFS. The flexibility of this framework is inherent in its ability to employ different subset-selection policies, which can lead, for example, to differing tradeoffs between mesh quality and complexity. In this work, we have advocated a particular subset-selection policy based on the ED scheme that yields the specific instance of GPRFS known as GPRFS-ED. Through experimental results, our GPRFS-ED method was shown to be superior to the state-of-the-art GPR scheme, yielding meshes of higher (in most cases) or comparable quality both in terms of PSNR and subjective quality, at much lower computational and memory costs. Furthermore, by making a different choice for the parameter γ in the GPRFS-ED method from the recommended nominal value of 4, one can easily tradeoff between computational/memory complexity and mesh quality. Herein, we have also proposed a reduced-complexity variant of the GPRFS-ED method, called GPRFS-MED, which can also be viewed as a modified version of the ED scheme. Our GPRFS-MED method was shown to produce vastly better quality meshes than ED scheme with only a reasonably modest increase in computational cost. Since our GPRFS-ED/GPRFS-MED method offers considerable flexibility in the tradeoff between mesh quality and complexity, the method is suitable for a wide range of applications with differing computational/memory constraints. Through the use of our newly proposed mesh-generation framework, GPRFS, and its related methods, GPRFS-ED and GPRFS-MED, one can obtain mesh-based image representations of higher quality, which benefits the many applications that utilize such representations. Furthermore, as future work, one might develop better subset-selection policies for use within our GPRFS framework, further enhancing its utility for the preceding applications.

REFERENCES

- [1] M. Sarkis and K. Diepold, "Content adaptive mesh representation of images using binary space partitions," *IEEE Trans. on Image Processing*, vol. 18, no. 5, pp. 1069–1079, May 2009.
- [2] 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.
- [3] L. Demaret and A. Iske, "Adaptive image approximation by linear splines over locally optimal Delaunay triangulations," *IEEE Signal Processing Letters*, vol. 13, no. 5, pp. 281–284, May 2006.
- [4] —, "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.
- [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] L. Demaret and A. Iske, "Scattered data coding in digital image compression," in *Curve and Surface Fitting: Saint-Malo 2002*. Brentwood, TN, USA: Nashboro Press, 2003, pp. 107–117.
- [7] 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.
- [8] N. Dyn, M. S. Floater, and A. Iske, "Adaptive thinning for bivariate scattered data," *Journal of Computational and Applied Mathematics*, vol. 145, pp. 505–517, 2002.
- [9] 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.
- [10] 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.
- [11] 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.
- [12] 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.
- [13] 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.
- [14] 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.
- [15] 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.
- [16] 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.
- [17] 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.

- [18] M. D. Adams, "An efficient progressive coding method for arbitrarily-sampled image data," *IEEE Signal Processing Letters*, vol. 15, pp. 629–632, 2008.
- [19] 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.
- [20] N. Dyn, D. Levin, and S. Rippa, "Data dependent triangulations for piecewise linear interpolant," *IMA Journal of Numerical Analysis*, vol. 10, pp. 137–154, 1990.
- [21] L. Darsa and B. Costa, "Multiresolution representation and reconstruction of adaptively sampled images," in *Proc. of SIBGRAPI*, 1996, pp. 321–328.
- [22] G. Wolberg, "Nonuniform image reconstruction using multilevel surface interpolation," in *Proc. of IEEE International Conference on Image Processing*, vol. 1, 1997, pp. 909–912.
- [23] Y. Eldar, M. Lindenbaum, M. Porat, and Y. Y. Zeevi, "The farthest point strategy for progressive image sampling," *IEEE Trans. on Image Processing*, vol. 6, no. 9, pp. 1305–1315, Sep. 1997.
- [24] M. Kashimura, Y. Sato, and S. Ozawa, "Image description for coding using triangular patch structure," in *Proc. of IEEE International Conference on Communications Systems*, Singapore, 1992, pp. 330–334.
- [25] M. Grundland, C. Gibbs, and N. A. Dogson, "Stylized multiresolution image representation," *Journal of Electronic Imaging*, vol. 17, no. 1, pp. 013 009.1–17, 2008.
- [26] D. Terzopoulos and M. Vasilescu, "Sampling and reconstruction with adaptive meshes," in *Proc. of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Maui, HI, USA, 1991, pp. 70–75.
- [27] L. Rila, "Image coding using irregular subsampling and Delaunay triangulation," in *Proc. of SIBGRAPI*, 1998, pp. 167–173.
- [28] Y. Yang, M. N. Wernick, and J. G. Brankov, "A fast approach for accurate content-adaptive mesh generation," *IEEE Trans. on Image Processing*, vol. 12, no. 8, pp. 866–881, Aug. 2003.
- [29] M. D. Adams, "An improved content-adaptive mesh-generation method for image representation," in *Proc. of IEEE International Conference on Image Processing*, Hong Kong, China, Sep. 2010, pp. 873–876.
- [30] H. Edelsbrunner, "Triangulations and meshes in computational geometry," *Acta Numerica*, pp. 133–213, 2000.
- [31] 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.
- [32] 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.
- [33] "USC-SIPI image database," 2010. [Online]. Available: <http://sipi.usc.edu/database>
- [34] "JPEG-2000 test images," ISO/IEC JTC 1/SC 29/WG 1 N 545, Jul. 1997.
- [35] R. W. Floyd and L. Steinberg, "An adaptive algorithm for spatial greyscale," *Proc. of the Society for Information Display*, vol. 17, no. 2, pp. 75–77, 1976.
- [36] M. Aubury and W. Luk, "Binomial filters," *Journal of VLSI Signal Processing*, vol. 12, pp. 35–50, 1996.
- [37] J. Zhang, S. Kamata, and Y. Ueshige, "A pseudo-Hilbert scan algorithm for arbitrarily-sized rectangle region," in *Advances in Machine Vision, Image Processing, and Pattern Analysis*, ser. Lecture Notes in Computer Science, vol. 4153. Springer-Verlag, 2006, pp. 290–299.
- [38] O. Devillers, "On deletion in Delaunay triangulations," in *ACM Annual Symposium on Computational Geometry*, Miami Beach, FL, USA, 1999, pp. 181–188.
- [39] "Kodak lossless true color image suite," 2010. [Online]. Available: <http://r0k.us/graphics/kodak>



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 in 1993, the M.A.Sc. degree in electrical engineering from the University of Victoria, Victoria, BC, Canada in 1998, and the Ph.D. degree in electrical engineering from the University of British Columbia, Vancouver, BC, Canada in 2002. From 1993 to 1995, Michael was a member of technical staff at Bell-Northern Research in Ottawa, ON, Canada where he developed real-time software for fiber-optic telecommunication systems. 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 (especially multirate and multiresolution signal processing), wavelets, data compression, computational geometry, computer graphics, and algorithms.