CrossMark

Anisotropic mesh adaptation for image representation

Xianping Li

Abstract

Triangular meshes have gained much interest in image representation and have been widely used in image processing. This paper introduces a framework of anisotropic mesh adaptation (AMA) methods to image representation and proposes a GPRAMA method that is based on AMA and greedy-point removal (GPR) scheme. Different than many other methods that triangulate sample points to form the mesh, the AMA methods start directly with a triangular mesh and then adapt the mesh based on a user-defined metric tensor to represent the image. The AMA methods have clear mathematical framework and provide flexibility for both image representation and image reconstruction. A mesh patching technique is developed for the implementation of the GPRAMA method, which leads to an improved version of the popular GPRFS-ED method. The GPRAMA method can achieve better quality than the GPRFS-ED method but with lower computational cost.

Keywords: Image representation, Adaptive sampling, Anisotropic mesh adaptation, Metric tensor, Mesh patching

1 Introduction

Triangular meshes have recently received considerable interest in adaptive sampling for image representation [1–17]. One common approach is to find proper sample points then connect the points to form a mesh. For example, Ramponi and Carrato [5] have defined a sample skewness parameter and used a multi-resolution approach to obtain a grid with an almost uniform sample density along the edges and no sample in areas with constant or linearly changing gray level. Yang et al. [8] argue that small (in area) elements are needed in image region where the second directional directive is large and have introduced the error diffusion (ED) scheme. They first construct a feature map based on the largest entry (of absolute value) in the Hessian matrix of the image function, then use Floyd-Steinberg dithering scheme to generate sample points, and finally use Delaunay triangulation to connect the nodes into a mesh. Demaret et al. [11, 12] have introduced the greedy-point removal (GPR) scheme that first constructs a triangular mesh using all the image points and then removes the sample points that yield smallest reconstruction error repeatedly. Adams [17] has proposed the GPRFS method based on the GPR scheme by replacing

the initial triangular mesh of all image points with a subset of the points and developed the GPRFS-ED method that selects the initial points using the ED scheme.

Another approach is to use a mesh directly to represent the image. For example, Terzopoulos and Vasilescu [2] have introduced an adaptive mesh approach where the mesh is considered as a dynamic node/spring system. They sample an image at a reduced rate and then reconstruct it by concentrating the nodes of the mesh at regions where the image values change rapidly (high-gradient region). They develop adaptive meshes with a feedback procedure that automatically adjusts spring parameters according to the observations made at the nodes to which they are attached, and use a Gaussian convolution of the Hessian for the adaptive image reconstruction. Isotropic triangles are used in their adaptive meshes. Courchesne et al. [13] use the Hessian matrix based on the gray level of MRI images as a metric tensor to adapt the triangular mesh for 3D reconstruction of human trunk. The Hessian matrix is reconstructed by linear or quadratic fitting. They then adapt the mesh based on the provided metric tensor and four constraint factors—minimum and maximum Euclidean edge lengths, maximum stretching of the metric, and target length of an edge in the metric. Bougleux et al. [15] have developed a progressive geodesic meshing algorithm that defines a geodesic distance using

Correspondence: lixianp@umkc.edu
Department of Mathematics and Statistics, the University of Missouri—Kansas City, 5120 Rockhill Rd, 64110 Kansas City, MO, USA

regularized Hessian as the metric tensor and exploits the anisotropy of images through a farthest point sampling strategy that forces the anisotropic Delaunay triangles to follow the geometry of the image. They have demonstrated the advantages of anisotropic triangular approximation over isotropic triangular approximation. Sarkis and Diepold [16] have used binary space partitions in combination with clustering scheme to approximate an image with a mesh. They first cluster the image into a few initial triangles (or rectangles) and then subdivide each triangle (or rectangle) into two or more smaller triangles recursively according to a predefined threshold.

Most of the adaptive sampling methods are "content-based" that use some information from the image such as edges, textures, or Hessian. Different sampling or meshing strategies have been developed. The GPR method provides high-quality meshes but requires significant computational cost. On the other hand, the ED method reduces computational cost but provides lower quality meshes. The GPRFS-ED method tries to find a balance between mesh quality and computational cost by combining the advantages of GPR and ED methods. It is worth mentioning that most of the sampling methods take the approach of finding the desired sample points first and then connect the points into a mesh. Only a few methods such as [2, 13, 15, 16] follow the approach that starts from an initial mesh and then adapt the mesh to represent the image.

On the other hand, anisotropic mesh adaptation (AMA) has been successfully applied to improve computational efficiency and accuracy when solving partial differential equations [18–23]. In this paper, we introduce a framework of AMA methods for image representation. AMA methods take the $\mathbb{M}$-uniform mesh approach for mesh adaptation and use finite element interpolation for image reconstruction. The methods start with an initial triangular mesh, then adapt the mesh according to a user-defined metric tensor $\mathbb{M}$, and finally reconstruct the image from the mesh. The framework has the flexibility for both mesh adaptation and image reconstruction. Various metric tensors can be chosen for mesh adaptation, and different orders of finite element interpolation can be applied for reconstruction. In this paper, we only consider linear finite element interpolation for triangular elements in the reconstruction step.

For reader's convenience, the representation methods under consideration are summarized in the following list.

- ED: error diffusion method developed by Yang et al. [8].
- AMA methods such as $\mathbb{M}_{aniso,k}$: anisotropic mesh adaptation method using metric tensor $\mathbb{M}_{aniso}$ with the initial mesh being adapted k times to generate the desired mesh, proposed in Section 3.

- GPR: greedy-point removal scheme proposed by Demaret et al. [11, 12].
- GPRFS-ED: modified GPR scheme starting from a subset of points chosen by ED method, proposed by Adams [17].
- GPRED-CDT(γ): GPR starting from γ times of the desired number of sample points chosen by ED and utilizing constrained Delaunay triangulation for mesh patching, essentially the same as GPRFS-ED, proposed in Section 4.
- GPRED-EC(γ): same as GPRED-CDT(γ) except using ear clipping for mesh patching, proposed in Section 4.
- GPRAMA(γ): GPR starting from an AMA representation of γ times of the desired sample density and utilizing ear clipping for mesh patching, proposed in Section 4.

The remainder of this paper is organized as follows. Firstly, a brief introduction of the AMA methods is given in Section 2 where the details of the methods can be found in [24–26]. Then in Section 3, the AMA representation framework is introduced and some results obtained from different methods are presented. In Section 4, a GPRAMA representation method based on AMA and GPR is proposed and some results and computational complexity are discussed. Finally, some conclusions and comments are given in Section 5. For reader's convenience, a brief summary of finite element interpolation for triangular elements is provided in the Appendix.

2 Anisotropic mesh adaptation (AMA) methods

Different adaptive sampling methods and mesh strategies have been applied in image representation by other researchers as summarized in Section 1. In this section, we introduce the "anisotropic mesh adaptation" (AMA) methods. AMA methods take the $\mathbb{M}$-uniform mesh approach, with which an adaptive mesh is generated as a uniform mesh in the metric specified by a tensor $\mathbb{M}$. The metric tensor $\mathbb{M}$ is required to be strictly positive definite and it determines the size, shape, and orientation of the triangular elements [25]. Once a metric tensor is specified, the free C++ code BAMG (bidimensional anisotropic mesh generator) developed by Hecht [27] is used to generate the corresponding triangular mesh. BAMG first generates an initial mesh based on the geometry of the domain provided in a file that defines the nodes and edges and the desired mesh size using constrained Delaunay triangulation. Then users have the choice to either provide a metric tensor on the initial background mesh or use the internal metric tensor computed by BAMG. Once a metric tensor $\mathbb{M}$ is provided, BAMG employs five local minimization tools including edge suppression, vertex suppression, vertex addition, edge swapping, and

vertex relocation to generate the desired anisotropic mesh according to $\mathbb{M}$. One of our objectives in this paper is to build the framework for AMA in image representation that can take different metric tensors for different needs in image processing.

Firstly, we introduce some notations and the conditions for $\mathbb{M}$-uniform meshes. Let Ω be the spatial domain, K be any triangular element in a simplicial mesh $\mathcal{T}_h$, and $\hat{K}$ to be the reference element that is equilateral and unitary in area. Let F_K be the affine mapping from $\hat{K}$ to K. An $\mathbb{M}$-uniform 2-D triangular mesh $\mathcal{T}_h$ for a given metric tensor $\mathbb{M} = \mathbb{M}(\mathbf{x})$ satisfies the following condition

$$(F_K')^T \mathbb{M}_K F_K' = \frac{\sigma_h}{N} I, \quad \forall K \in \mathcal{T}_h \tag{1}$$

that is equivalent to the following two conditions [25]

$$|K|\sqrt{\det(\mathbb{M}_K)} = \frac{\sigma_h}{N}, \quad \forall K \in \mathcal{T}_h, \tag{2}$$

$$\frac{1}{2}\mathrm{tr}\left((F_K')^T \mathbb{M}_K F_K'\right) = \det\left((F_K')^T \mathbb{M}_K F_K'\right)^{\frac{1}{2}}, \quad \forall K \in \mathcal{T}_h, \tag{3}$$

where I is the identity matrix of size 2×2, $|K|$ is the area of the element K, N is the number of mesh elements, F_K' is the Jacobian matrix of F_K,

$$\mathbb{M}_K = \frac{1}{|K|} \int_K \mathbb{M}(\mathbf{x}) d\mathbf{x}, \text{ and } \sigma_h = \sum_{K \in \mathcal{T}_h} |K|\sqrt{\det(\mathbb{M}_K)}. \tag{4}$$

Condition (2) is called the *equidistribution condition* and determines the size of element K, while condition (3) is called the *alignment condition* and characterizes the shape and orientation of K.

In the framework of AMA methods, the goal is to develop and use proper metric tensors based on the needs of the problems. Different metric tensors will have different properties and features. It is worth mentioning that Hessian matrix H is not an optimal metric tensor [26] and may not be positive definite. In our framework, we replace the Hessian with its absolute form defined as follows

$$|H| = Q \begin{bmatrix} |\lambda_1| & 0 \\ 0 & |\lambda_2| \end{bmatrix} Q^{-1}, \text{ with } H = Q \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} Q^{-1}, \tag{5}$$

where λ_1 and λ_2 are the eigenvalues of H and Q is the matrix of the corresponding eigenvectors. The metric tensor $|H|$ is denoted as $\mathbb{M}_H$ in this paper, and [13] can be considered as a specific example in our AMA framework. Some other metric tensors are described below.

For isotropic mesh adaptation, a metric tensor $\mathbb{M}_{iso}$ is defined for any triangular element K as follows [24]

$$\mathbb{M}_{iso,K} = \left(1 + \frac{1}{\alpha_h}\|H_K\|_F\right) I, \tag{6}$$

where H_K denotes the value of H at the center of element K, $\|\cdot\|_F$ is the Frobenius matrix norm, and α_h is a regularization factor that is defined by

$$\alpha_h = \frac{1}{|\Omega|} \left(\sum_{K \in \mathcal{T}_h} |K| \cdot \|H_K\|_F \right). \tag{7}$$

$\mathbb{M}_{iso}$ provides isotropic mesh adaptation where all triangles are of the same shape but may have different sizes, and more triangles will be concentrated in the high-gradient region.

For anisotropic mesh adaptation, a metric tensor $\mathbb{M}_{aniso}$ is developed in [24] that is based on minimization of a bound on the H^1 semi-norm of linear interpolation error and is defined for any triangular element K as follows

$$\mathbb{M}_{aniso,K} = \rho_K \det\left(I + \frac{1}{\alpha_h}|H_K|\right)^{-\frac{1}{2}} \left[I + \frac{1}{\alpha_h}|H_K|\right], \tag{8}$$

and

$$\rho_K = \left\|I + \frac{1}{\alpha_h}|H_K|\right\|_F^{\frac{1}{2}} \det\left(I + \frac{1}{\alpha_h}|H_K|\right)^{\frac{1}{4}}, \tag{9}$$

where α_h is the regularization parameter and is defined implicitly through

$$\sum_{K \in \mathcal{T}_h} \rho_K |K| = 2|\Omega|. \tag{10}$$

With this choice of α_h, roughly 50 % of the triangular elements will be concentrated in large gradient regions [24]. The adaptation is anisotropic because the triangles in the mesh may have different size, shape, and orientation.

For image processing with anisotropic diffusion filters [28], a metric tensor $\mathbb{M}_{DMP}$ is developed in [20] that takes the inverse of the diffusion tensor. The elements of the mesh based on $\mathbb{M}_{DMP}$ will be aligned along the principle diffusion direction, and the corresponding numerical solution will satisfy the maximum principle under some conditions of time step [22]. Another metric tensor $\mathbb{M}_{DMP+adap}$ is also developed in [20] that combines the properties of both $\mathbb{M}_{aniso}$ and $\mathbb{M}_{DMP}$, that is, the mesh not only provides numerical solution that satisfies maximum principle but also performs adaptation based on the interpolation error.

In this paper, we only focus on the metric tensors $\mathbb{M}_H, \mathbb{M}_{iso}$, and $\mathbb{M}_{aniso}$. In the computations for those metric tensors, the Hessian matrix H at a point is reconstructed by the least-squares fitting from function values at neighboring vertices. For convenience, we use the metric tensor to denote the mesh as well as the corresponding representation. For example, $\mathbb{M}_{aniso}$ denotes the mesh and representation according to the metric tensor $\mathbb{M}_{aniso}$.

As demonstrated in [15, 26], anisotropic meshes have advantages over isotropic meshes in terms of computational efficiency and accuracy. Our results in Section 3 also confirm that anisotropic meshes provide better representation quality than isotropic meshes. Therefore, we will only use anisotropic mesh adaptation methods in the AMA image representation framework.

3 AMA image representation framework

In this section, we introduce the AMA framework for image representation. We consider an image as a function f that is defined on a set Λ of points on domain $\Omega = [0, 1] \times [0, 1]$. Let S denote the set of desired sample points and SD denote the sample density that is defined as

$$SD = |S|/|\Lambda|, \tag{11}$$

where $|\cdot|$ is the cardinality of the set. The quality of the mesh (or representation) is measured by the peak-signal-to-noise-ratio (PSNR) that is calculated in decibels (dB) as follows [17]

$$\text{PSNR} = 20 \log_{10} \left(\frac{2^p - 1}{d} \right), d = \left(\frac{1}{|\Lambda|} \sum_{i \in \Lambda} |\hat{f}(i) - f(i)|^2 \right)^{\frac{1}{2}}, \tag{12}$$

where $\hat{f}$ is the reconstructed image from the triangular mesh, and p is the sample precision in bits/sample. Larger value of PSNR indicates better mesh quality (or representation).

3.1 Framework

In this paper, we apply the AMA methods for image representation, in which a triangular mesh with fewer points is used to represent the original image f. Given the original image f, we use the free C++ code BAMG [27] to generate an initial triangular mesh with number of vertices N_v that is much smaller than $|\Lambda|$. Then the values on the vertices are interpolated from f, and the metric tensor $\mathbb{M}$ is computed for each triangle in the initial mesh. With the computed metric tensor, BAMG generates the desired anisotropic mesh using the procedures described in Section 2. Finally, the image is reconstructed from the final mesh using finite element interpolation.

It is difficult to generate a mesh that satisfies conditions (2) and (3) exactly. Moreover, the initial mesh, in general, may not contain the important information from the original image. Therefore, the mesh can be adapted multiple times in order to obtain a final mesh that is close to be an $\mathbb{M}$-uniform mesh, or the so-called quasi-$\mathbb{M}$-uniform mesh. The iteration can be terminated if further adaptation does not significantly improve image quality (measured by PSNR) and can be image dependent in order to obtain the best representation of a particular image. However, numerical results show that 2 to 5 iterations are sufficient to provide a quasi-$\mathbb{M}$-uniform mesh with good quality, and further adaptation does not improve the quality significantly.

More specifically, the AMA representation framework consists of the following four steps.

Step 1: Generate an initial mesh based on the desired sample density.

Step 2: Assign function values to mesh vertices (and interpolation nodes) from original image using linear finite element interpolation and compute the user-defined metric tensor $\mathbb{M}$ on the mesh.
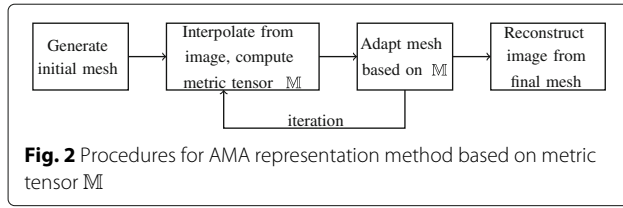
Step 3: Adapt the mesh to be a quasi-$\mathbb{M}$-uniform mesh that almost fits the provided metric tensor $\mathbb{M}$.

Step 4: Reconstruct the image using the final quasi-$\mathbb{M}$-uniform mesh with finite element interpolation for triangles.

During the reconstruction step (step 4), for a particular image pixel, we first locate the triangle that the pixel lies on or in. Then, we compute the coordinates of the pixel in the reference element (see Fig. 1) and the corresponding basis functions at the interpolation nodes. Finally, we interpolate the function value from the interpolation nodes using the basis functions as the weights. For linear interpolation, only the three vertices are needed, while for quadratic interpolation, the midpoints are also needed. For reader's convenience, a brief summary of
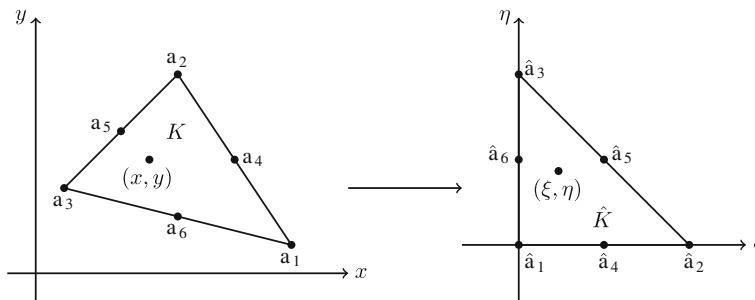
Fig. 1 Sketch of *triangular* element K and its reference element $\hat{K}$ for finite element interpolation, where $\hat{K}$ is an isosceles *right triangle* with vertices $\hat{\mathbf{a}}_1 (0, 0)$, $\hat{\mathbf{a}}_2 (1, 0)$, and $\hat{\mathbf{a}}_3 (0, 1)$

Fig. 2 Procedures for AMA representation method based on metric tensor $\mathbb{M}$

finite element interpolation for triangular elements is provided in the Appendix.

The above procedures are shown in Fig. 2 where steps 2 and 3 can be repeated multiple times in order to obtain better results. For convenience, we denote the number of iterations for steps 2 and 3 by k and the corresponding mesh as $\mathbb{M}_k$. For example, for metric tensor $\mathbb{M}_H$, the representation is denoted as $\mathbb{M}_{H,k}$ if there are k iterations of steps 2 and 3. For metric tensor $\mathbb{M}_{\text{aniso}}$, the corresponding mesh is denoted as $\mathbb{M}_{\text{aniso},k}$. When $k = 1$, the mesh is only adapted from initial mesh once and no further adaptation is performed. As mentioned before, we take $k \in [2, 5]$ in our computations.

Note that we can start with a random initial mesh that has more number of vertices than desired ($N_v > |S|$) in Step 1, then iterate steps 2 and 3 to obtain a mesh with desired sample density. By this way, more information from the original image could be reserved by the mesh. Another approach is to obtain an initial mesh with $N_v > |S|$ for steps 1 to 3, then reduce the number of mesh vertices to the desired number $|S|$ using the GPR algorithm before Step 4. The particular representation using GPR before step 4 is denoted as GPRAMA and will be discussed in Section 4.

Note that in step 4, we can choose different orders of finite element interpolation methods. However, in this paper, we only consider linear finite element interpolation for triangular elements and the effects of higher order

interpolation on representation quality is currently under investigation. In fact, quadratical interpolation provides higher representation quality; however, the sample density is also higher since it uses the midpoints on the edges of the triangles. There is no need to sample the midpoints because their coordinates can be computed; however, the function values at the midpoints need to be assigned in Step 2. A fair comparison is needed between higher order interpolation and linear interpolation with the same sample density.

3.2 Results

For evaluation purpose, we take the two widely used images, "Lena" and "peppers," available from USC-SIPI Image Database [29]. Figure 3 shows the initial images of Lena and peppers with pixel resolution 512×512, while the RGB components of each pixel are converted to greyscale luminance using the weighted sum $0.2989 \cdot R + 0.5870 \cdot G + 0.1140 \cdot B$. Three more images with different resolutions and features are also tested for comparison purpose, including "roof," "lighthouse," and "saturn." Image roof has resolution 1024×1024 and is obtained from USC-SIPI Image Database [29]. "lighthouse" has resolution 480×640 and "saturn" has resolution 1500×1200; both are taken from MATLAB R2016a image data folder. All images are converted to greyscale images as done for images Lena and peppers.

Figure 4 shows the representation of the image Lena at $SD = 3\ \%$ using isotropic mesh according to $\mathbb{M}_{\text{iso},3}$, and the quality of representation is PSNR $= 28.26$. Figure 5 shows the $\mathbb{M}_{\text{aniso},1}$ and $\mathbb{M}_{\text{aniso},2}$ meshes and the corresponding sample points of the image Lena at $SD = 3\ \%$. The representation quality for $\mathbb{M}_{\text{aniso},1}$ is PSNR $= 29.81$, and is PSNR $= 30.81$ for $\mathbb{M}_{\text{aniso},2}$. After three iterations, the quality increases to PSNR $= 31.00$ for $\mathbb{M}_{\text{aniso},3}$ as shown in Fig. 6. Further adaptation does not

Fig. 3 Images from USC-SIPI Image Database [29] with pixel resolution 512×512: **a** Lena; **b** peppers. The RGB components of each pixel are converted to luminance using the weighted sum $0.2989 \cdot R + 0.5870 \cdot G + 0.1140 \cdot B$
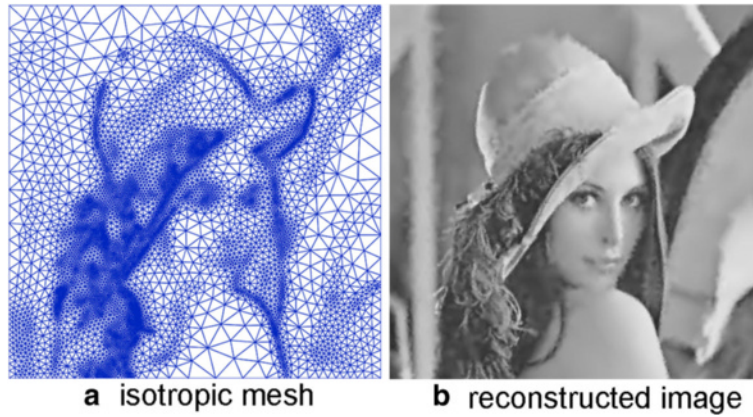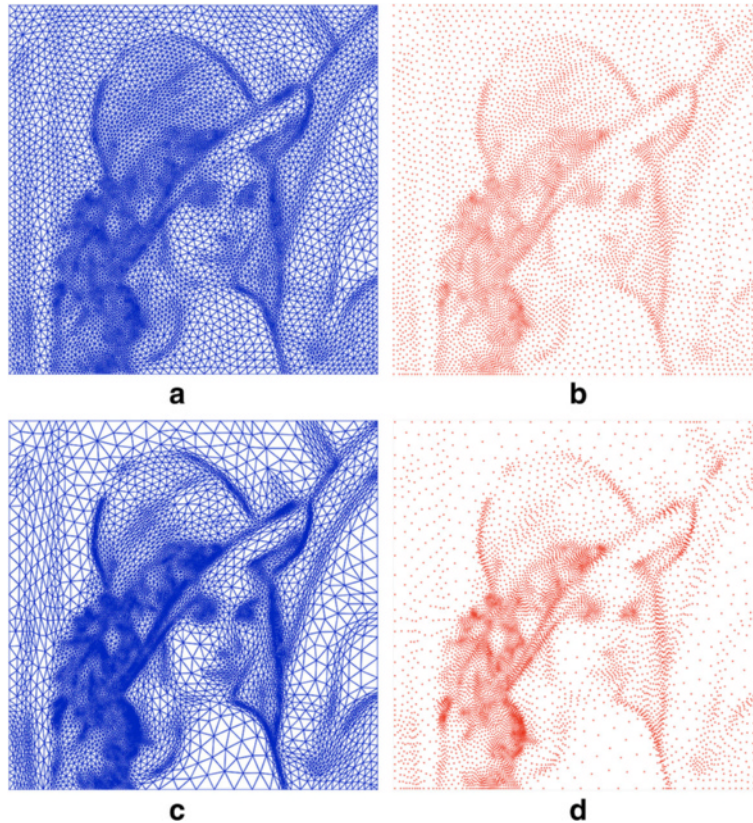
Fig. 4 Representation of image Lena at sample density of 3 % using $\mathbb{M}_{iso,3}$: **a** *triangular* mesh $\mathbb{M}_{iso,3}$; **b** reconstructed image, PSNR=28.47

improve the representation quality for this case. It is clear that image representation based on $\mathbb{M}_{aniso}$ is better than the one based on $\mathbb{M}_{iso}$.

Figure 6 also shows the representation and reconstruction of the image Lena at $SD = 3$ % using ED scheme denoted as ED. For the ED scheme, we have applied the strategies recommended in [17], including

$B(3)$ smoothing for image data, zero extension for boundary points, and serpentine scan order for the error diffusion. The representation quality is PSNR=28.41 for ED.

Figure 7 shows similar results for the image peppers at $SD = 3$ %. The quality for $\mathbb{M}_{aniso,3}$ is PSNR=30.89 and is PSNR=28.05 for ED. As can be seen from Figs. 6 and 7, the $\mathbb{M}_{aniso,3}$ meshes preserve key features of the

Fig. 5 Meshes and sample points of image Lena at sample density of 3 % using $\mathbb{M}_{aniso}$: **a** $\mathbb{M}_{aniso,1}$ mesh; **b** sample points from the mesh in **a**; **c** $\mathbb{M}_{aniso,2}$ mesh; **d** sample points from the mesh in **c**
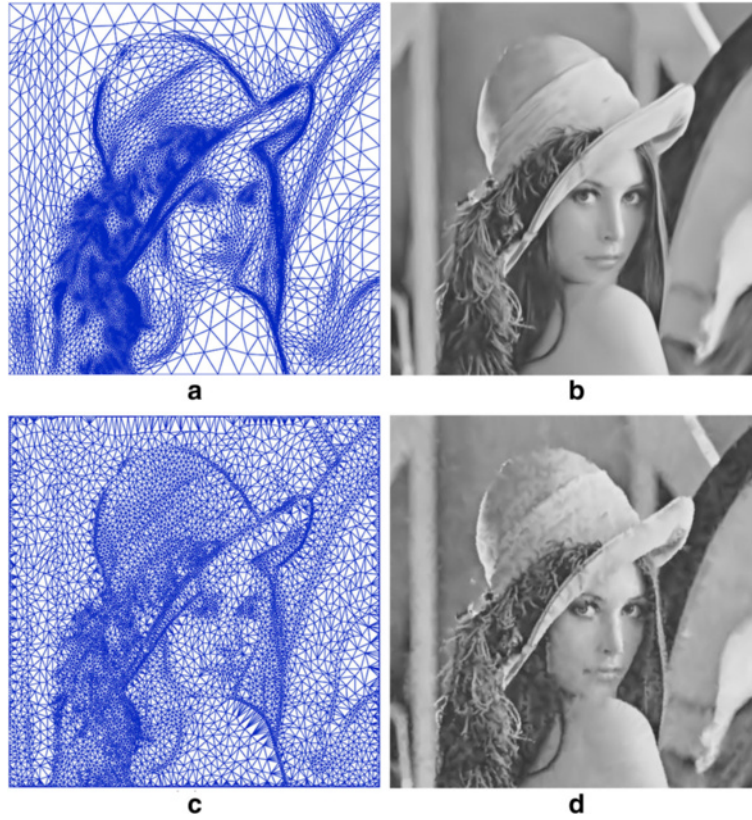
Fig. 6 Representations of image Lena at sample density of 3 % using different methods: **a** *triangular* mesh $\mathbb{M}_{\text{aniso},3}$; **b** reconstructed image from **a**, PSNR=31.00; **c** mesh obtained using ED scheme; **d** reconstructed image from **c**, PSNR=28.41

original image by concentrating more triangular elements around the edges and texture regions. The quality of $\mathbb{M}_{\text{aniso},3}$ representation is much better than that of the ED representation.

The representation qualities for both image Lena and peppers using different AMA methods are shown in Table 1. The results obtained using ED scheme are also presented for comparison purpose. As can be seen, the quality of $\mathbb{M}_{\text{iso},3}$ representation is comparable to ED but not as good as the anisotropic ones.

The results confirm that more iterations of steps 2 and 3, that is, increasing the values of k, do improve the mesh quality. For example, for image Lena at $SD = 3$ %, PSNR increases from 29.67 for $\mathbb{M}_{H,1}$ to 30.45 for $\mathbb{M}_{H,3}$ and from 29.81 for $\mathbb{M}_{\text{aniso},1}$ to 31.00 for $\mathbb{M}_{\text{aniso},3}$. Similar results are observed for image peppers. The reason is that after each mesh adaptation, better information are preserved from the original image and the mesh is closer to the $\mathbb{M}$-uniform mesh. However, for the two images we are investigating, $k = 3$ already provides a mesh with good quality and further adaptation does not make significant improvement. The results of PSNR values using $\mathbb{M}_{H,k}$ and $\mathbb{M}_{\text{aniso},k}$ at different k values are presented in Table 2 for different images at sample density of 3 %. The optimal value of k depends on the given image. For example, $k = 2$ is the best for image saturn, $k = 3$ works the best for images Lena, and peppers. For image lighthouse, $\mathbb{M}_{H,k}$ and $\mathbb{M}_{\text{aniso},k}$ have different optimal k values and $k = 3$ is a good balance.

Comparing $\mathbb{M}_{H,1}$ and $\mathbb{M}_{\text{aniso},1}$ in Table 1 and 2, we see that the absolute Hessian $|H|$ is not an optimal metric tensor, and the performance of $\mathbb{M}_{\text{aniso},1}$ is better than $\mathbb{M}_{H,1}$ for all cases, although the difference is not significant. By adapting both $\mathbb{M}_H$ and $\mathbb{M}_{\text{aniso}}$ meshes three times, $\mathbb{M}_{\text{aniso},3}$ performs better than $\mathbb{M}_{H,3}$ for image Lena while the opposite occurs for image peppers. The results of $\mathbb{M}_{H,3}$ and $\mathbb{M}_{\text{aniso},3}$ for the other three images are presented in Table 3. The qualities of the representation depend on the specific image but overall performance is comparable for $\mathbb{M}_H$ and $\mathbb{M}_{\text{aniso}}$. In this paper, we choose $\mathbb{M}_{\text{aniso},3}$ as the representative from the AMA framework and propose a new image representation method based on AMA and GPR in the next section.
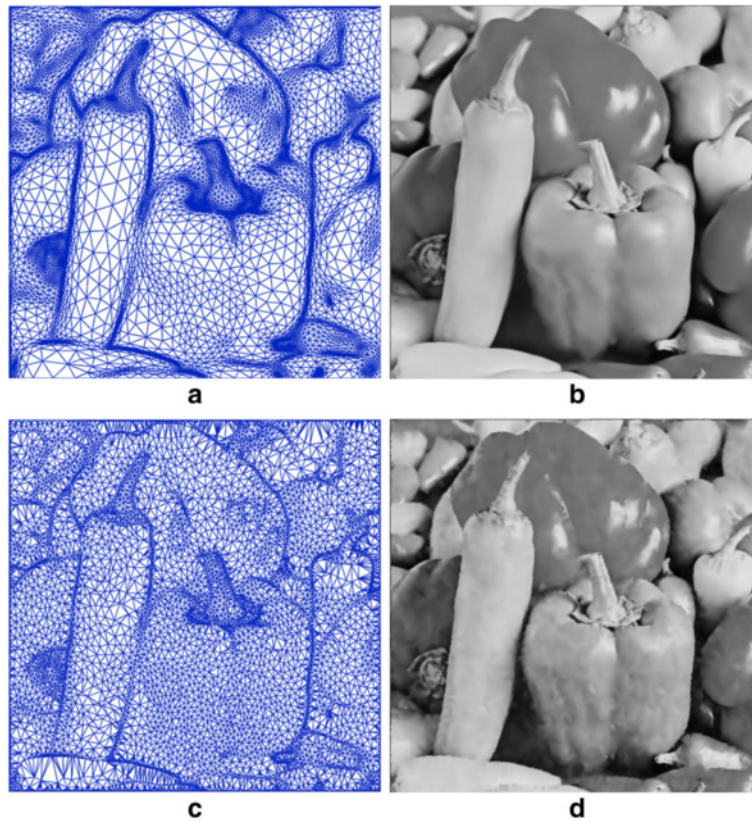
Fig. 7 Representations of image peppers at sample density of 3 % using different methods: **a** *triangular* mesh $\mathbb{M}_{aniso,3}$; **b** reconstructed image from **a**, PSNR=30.89; **c** mesh obtained using ED scheme; **d** reconstructed image from **c**, PSNR=28.05

4 GPRAMA representation method

In this section, we apply the greedy-point removal scheme to AMA representation and propose a new method denoted as GPRAMA. Adams has proposed the GPRFS method in [17] that is based on the GPR scheme while replacing the initial triangular mesh of all image points with a subset $S_0 \subseteq \Lambda$. The GPRFS method starts with $|S_0| = \gamma |S|$ for $\gamma \in [4, 5.5]$ and then uses GPR scheme to reduce the number of points from $|S_0|$ to the desired number $|S|$. Adams employs ED method to choose S_0 and denotes the method as GPRFS-ED in his paper.

As discussed in the previous section, our AMA representation methods provide better quality than ED scheme. Therefore, it is reasonable to consider the vertices of an AMA mesh as the initial subset S_0 for the GPR scheme. In this sense, the GPRAMA method is a specific example of the GPRFS method. However, GPRFS method relies on Delaunay triangulation of the sample points that does not have the anisotropic feature as in AMA meshes. In fact, the mesh quality obtained via Delaunay triangulation may not be optimal for a given set of sample points, measured by PSNR value of the reconstructed image. Figure 8 shows one example, where the mesh is obtained by Delaunay triangulation of the provided sample points and the mesh

quality is PSNR=28.57; however, for the same set of sample points, the $\mathbb{M}_{aniso,3}$ mesh (see Fig. 6a) has better quality with PSNR=31.00.

4.1 Mesh patching technique

In order to preserve the anisotropic feature of the AMA meshes when applying GPR scheme, we have developed a mesh patching technique that attaches a local mesh to the AMA mesh. This patching technique also works for any other triangular meshes. Let i be the index of a general vertex in the mesh, we denote the region covered by triangles sharing vertex i including the boundary edges by $\overline{\omega_i}$ and call it the patch of i. The polygon surrounding i formed by the boundary edges, is denoted as $\partial \omega_i$, and the inner region of the patch, that is, excluding the boundary edges, is denoted as ω_i. Before removing the vertex i, the patch $\overline{\omega_i}$ is partitioned by the triangular elements from the initial mesh, and linear finite element interpolation are used on those triangles for image reconstruction. If vertex i is chosen to be removed, $\overline{\omega_i}$ is triangulated again without vertex i. The new triangulation of $\overline{\omega_i}$ is then added to the global mesh structure. Figure 9 provides an illustration of the mesh patching technique with two different triangulation

Table 1 Comparison of mesh qualities obtained with various methods

Image	Sample Density (%)	PSNR (dB)					
		ED	$\mathbb{M}_{iso,3}$	$\mathbb{M}_{H,1}$	$\mathbb{M}_{aniso,1}$	$\mathbb{M}_{H,3}$	$\mathbb{M}_{aniso,3}$
Lena	1.0	21.38	24.13	25.42	25.57	26.32	*26.51*
	2.0	26.36	26.55	28.08	28.38	28.93	*29.35*
	3.0	28.41	28.26	29.67	29.81	30.45	*31.00*
	4.0	29.77	29.42	30.71	30.83	31.82	*31.99*
	6.0	31.46	31.06	32.09	32.26	33.22	*33.31*
Peppers	1.0	20.66	23.50	25.01	25.69	*26.31*	25.80
	2.0	25.30	26.10	28.16	28.55	29.34	*29.36*
	3.0	28.05	28.31	29.65	30.01	*31.05*	30.89
	4.0	29.57	29.14	30.74	30.94	*31.86*	31.86
	6.0	31.03	31.11	31.90	32.01	*33.10*	32.95

methods of the patch—one is the constrained Delaunay triangulation (CDT) [30] and the other is the ear clipping (EC) method [31]. Both CDT and EC work well for general polygons including concave ones. The specific procedures for the mesh patching of $\overline{\omega_i}$ are as follows.

Step 1: Save the global indices of the vertices of the polygon $\partial \omega_i$.

Step 2: Triangulate $\overline{\omega_i}$ without vertex i (using CDT or EC) and save the mesh connectivity that lists the local indices of the vertices of each triangle.

Step 3: Map the local indices in Step 2 to the global indices from Step 1, and add the updated connectivity information of $\overline{\omega_i}$ to the global mesh.

4.2 GPRAMA method

With the mesh patching technique described above and the AMA representation framework introduced in Section 3, the GPRAMA method consists of the following four steps.

Table 2 PSNR (dB) of $\mathbb{M}_{H,k}$ and $\mathbb{M}_{aniso,k}$ representations at sample density of 3 %

Image	Mesh	PSNR (dB)				
		$k=1$	$k=2$	$k=3$	$k=4$	$k=5$
Lena	$\mathbb{M}_{H,k}$	29.67	29.97	*30.45*	30.19	30.18
	$\mathbb{M}_{aniso,k}$	29.81	30.48	*31.00*	30.01	29.80
Lighthouse	$\mathbb{M}_{H,k}$	26.54	26.68	26.76	*26.85*	26.68
	$\mathbb{M}_{aniso,k}$	26.69	*26.85*	26.78	26.62	26.19
Peppers	$\mathbb{M}_{H,k}$	29.65	30.16	*31.05*	29.93	29.48
	$\mathbb{M}_{aniso,k}$	30.01	29.74	*30.89*	29.92	29.69
Roof	$\mathbb{M}_{H,k}$	27.40	28.07	28.43	28.66	*28.79*
	$\mathbb{M}_{aniso,k}$	27.55	28.16	28.48	28.62	*28.79*
Saturn	$\mathbb{M}_{H,k}$	48.89	*49.93*	48.74	46.05	45.91
	$\mathbb{M}_{aniso,k}$	48.97	*49.89*	49.77	48.87	47.91

Table 3 Comparison of mesh qualities obtained with $\mathbb{M}_{H,3}$ and $\mathbb{M}_{aniso,3}$

Image	Sample density (%)	PSNR (dB)		
		ED	$\mathbb{M}_{H,3}$	$\mathbb{M}_{aniso,3}$
Lighthouse	1.0	20.16	23.49	*23.51*
	2.0	24.10	25.40	*25.58*
	3.0	25.69	26.76	*26.78*
	4.0	26.79	27.69	*27.84*
	6.0	28.49	29.15	*29.22*
Roof	1.0	19.19	24.90	*25.31*
	2.0	23.39	27.07	*27.42*
	3.0	26.61	28.43	*28.48*
	4.0	28.02	*29.33*	29.33
	6.0	29.24	*30.41*	30.38
Saturn	1.0	42.14	47.72	*47.99*
	2.0	46.18	49.07	*49.32*
	3.0	47.33	48.74	*49.77*
	4.0	48.18	48.94	*49.74*
	6.0	48.48	*47.92*	47.74

Step 1: Generate an AMA representation based on a metric tensor $\mathbb{M}$ with number of vertices $N_v = \gamma |S|$ for $\gamma \geq 1$. Let V be the set of all vertices in the mesh and let $V_p = V$.

Step 2: For any vertex $v_i \in V_p$, compute the significance measure δe_i defined as the difference between the local mean square error of $\overline{\omega_i}$ after and before removing v_i, as shown below.

$$\delta e_i = \sum_{j \in \Lambda \cap \overline{\omega_i}} |\hat{f}_a(j) - f(j)|^2 - \sum_{j \in \Lambda \cap \overline{\omega_i}} |\hat{f}_b(j) - f(j)|^2, \quad (13)$$

where $\hat{f}_a$ is the reconstructed value using the new triangulation of the patch without vertex i and $\hat{f}_b$ is the one with vertex i.

Step 3: For the vertex $v_i \in V$ having minimal δe_i, reset V_p as the set of vertices of the polygon $\partial \omega_i$, remove v_i and apply the mesh patching technique for $\overline{\omega_i}$. Reset $V = V \backslash \{v_i\}$ and $N_v = N_v - 1$.

Step 4: If $N_v \leq |S|$, output the mesh and stop; otherwise, go to step 2 with the updated V_p and V.

In our computations, we choose metric tensor $\mathbb{M}_{aniso,3}$ in the above procedures for GPRAMA due to its good representation quality as described in Section 3. In each iteration of Step 2, except the first, we only need to compute the significance measures for the neighboring vertices of v_i after it is removed. Other vertices in the mesh outside of $\overline{\omega_i}$ are not affected. For efficient implementation, we do not need to delete the information at vertex v_i such as coordinates, function value and neighboring
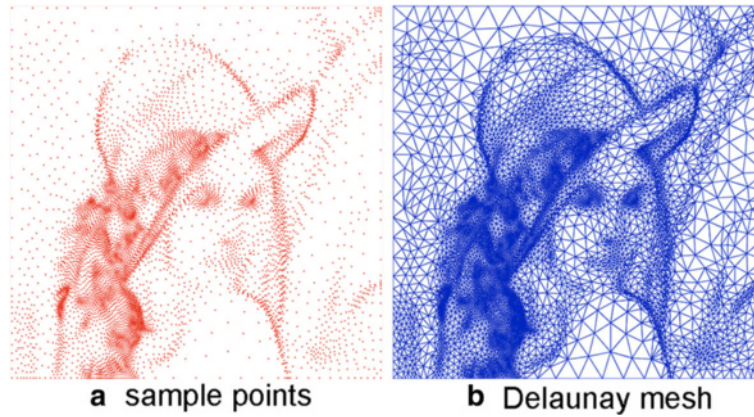
Fig. 8 Representation of image Lena at sample density of 3 %: **a** sample points; **b** Delaunay mesh, PSNR=28.57

triangles from the mesh data then triangulate the patch using the updated mesh data. We just need to replace the old triangulation of the patch containing v_i with the new one without v_i by updating the corresponding entries in the mesh data directly. A sorted index array for the significance measures can be used in step 3 for efficient selection of the vertex to be removed.

4.3 Results

According to the different triangulation methods for the mesh patching technique in step 3, the final mesh and the corresponding representation are denoted as GPRAMA(γ)-CDT if constrained Delaunay triangulation is used and GPRAMA(γ)-EC if ear clipping method is used, where γ specifies the number of initial points
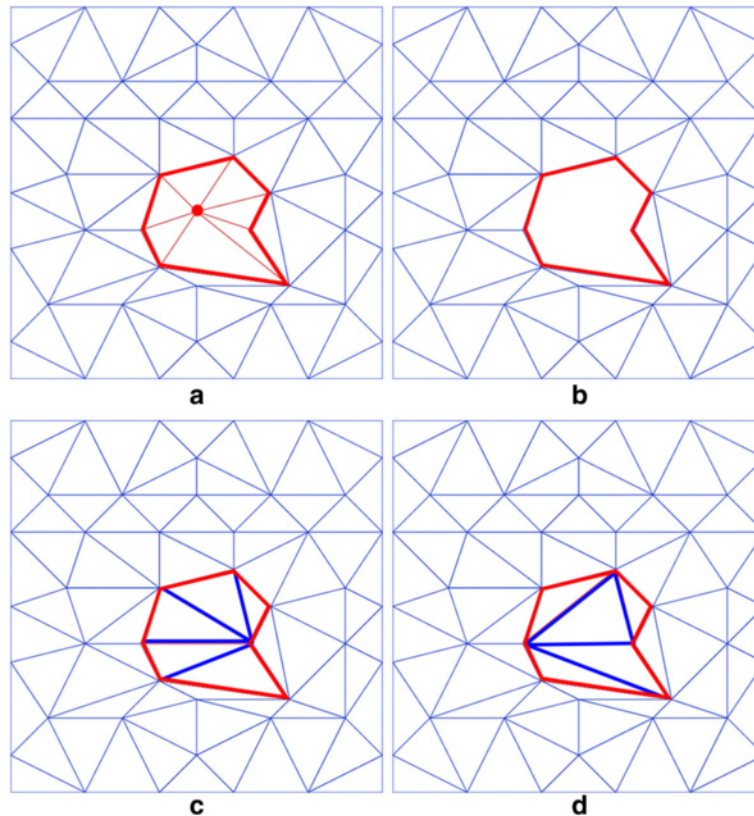
Fig. 9 Illustration of mesh patching technique: **a** original mesh with the highlighted patch of the point to be removed; **b** mesh with polygon of the empty patch; **c** mesh patching using constrained Delaunay triangulation (CDT) for **b**; **d** mesh patching using ear clipping (EC) method for **b**
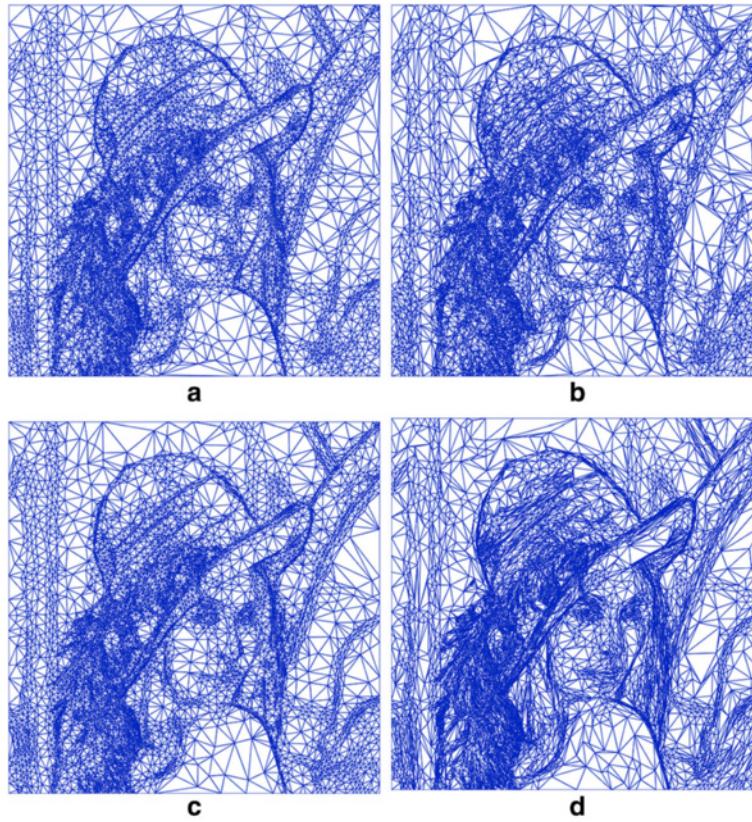
Fig. 10 Meshes of different GPR-related representations for image Lena at sample density of 3 %: **a** GPRED(4)-CDT, PSNR=33.49; **b** GPRED(4)-EC, PSNR=33.85; **c** GPRAMA(4)-CDT, PSNR=33.18; **d** GPRAMA(4)-EC, PSNR=34.51

$|S_0| = \gamma|S|$. Figure 10 shows the meshes obtained using GPRED and GPRAMA methods for image Lena at sample density 3 % with $\gamma = 4$, and Fig. 11 presents two of the reconstructed images.

GPRED(4)-CDT has quality PSNR=33.49 while GPRED(4)-EC has PSNR=33.85. GPRAMA(4)-CDT has quality PSNR=33.18 while GPRAMA(4)-EC has PSNR=34.51. The GPRED-CDT method is essentially the GPRFS-ED method in Adam's paper [17] while GPRED-EC is an improved version of GPRED-CDT due to the different triangulation of the local patch. For GPRAMA, using constrained Delaunay triangulation

Fig. 11 Reconstructed images using different GPR-related representations for image Lena at sample density of 3 %: **a** GPRED(4)-EC, PSNR=33.85; **b** GPRAMA(4)-EC, PSNR=34.51

for mesh patching does not preserve the anisotropy of the initial mesh, especially when significant amount of points are removed. Therefore, ear clipping method works better for GPRAMA, and GPRAMA-EC gives the best quality among the four GPR-related representations. Similar results are observed for image peppers as shown in Figs. 12 and 13.

The mesh qualities at different sample densities for both Lena and peppers using GPR-related representations are shown in Table 4, where mesh patching with ear clipping is used for GPR and GPRAMA methods. GPRAMA(4) provides better quality than the traditional GPR method for both images except at sample density of 1 % for image peppers. Furthermore, GPRAMA(3) performs better than GPRED(5)-CDT and is comparable with GPRED(5)-EC for both images. Therefore, GPRAMA can achieve better quality than GPRED while starting with smaller $|S_0|$ which indicates less computational cost, especially for high-resolution images. The computational costs for different methods are provided in the next subsection.

Figure 14 shows a picture of the Golden Gate Bridge of pixel resolution 4000×3000 and its GPRAMA representation at sample density 1 % with $\gamma = 4$. The reconstructed

image is shown in Fig. 15 and the representation quality is PSNR=34.30. Some results using different representation methods are provided in Table 5, where ear clipping is used in the mesh patching for all the GPR-related methods. As can be seen, GPRAMA(2) has quality as good as GPRED(4) but starts with only half of the initial points. The results are consistent with our previous observations for other images.
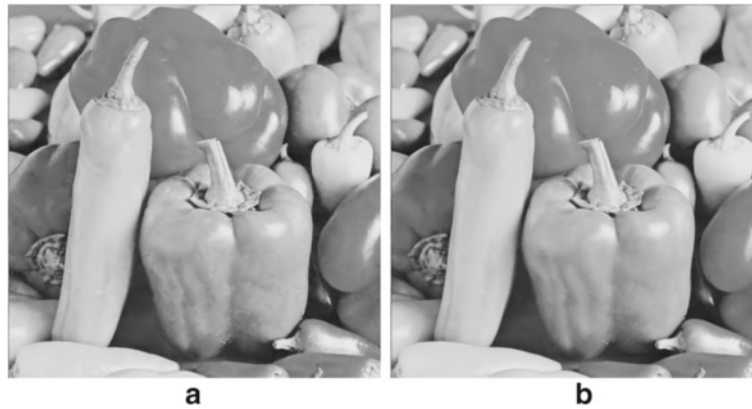
4.4 Computational complexity

Here, we compare the computational complexity of the various image representation methods considered in this paper. The computational complexity is measured in terms of CPU execution time in seconds (converted from clock ticks) and varies for different hardware and software environment. Our computations in this paper are performed in a MacBook Pro laptop with 2.6 GHz Intel Core i7 CPU, 8GB 1600MHz DDR3 memory, and OSX 10.9.5 operating system.

Note that our program code was developed with basic level of efficiency and has not been optimized for execution speed. Thus, the absolute CPU time for each method may be reduced by using highly optimized code. However,
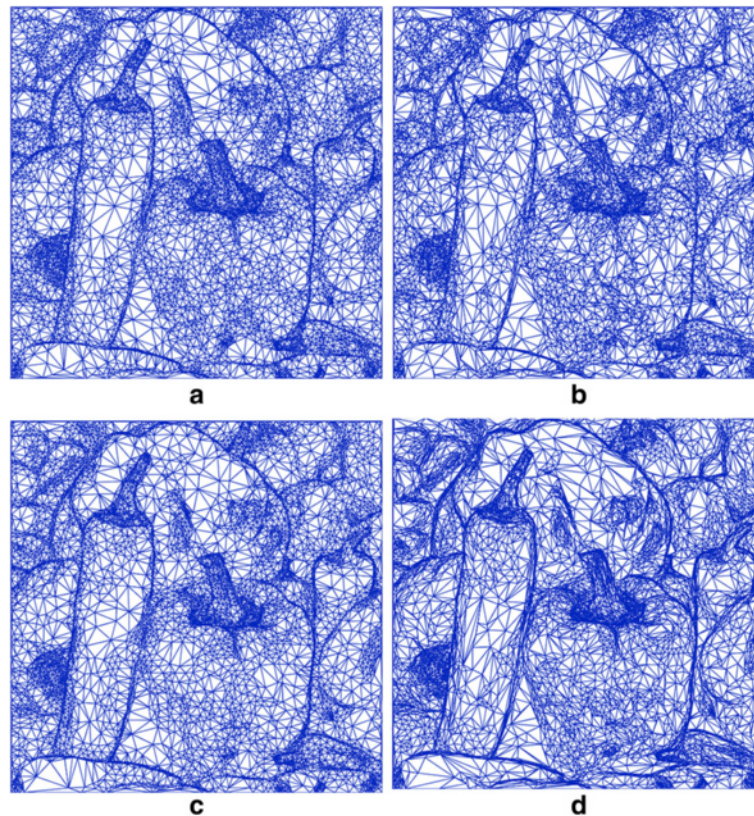
Fig. 12 Meshes of different GPR-related representations for image peppers at sample density of 3 %: **a** GPRED(4)-CDT, PSNR=33.65; **b** GPRED(4)-EC, PSNR=33.74; **c** GPRAMA(4)-CDT, PSNR=33.44; **d** GPRAMA(4)-EC, PSNR=34.23

Fig. 13 Reconstructed images using different GPR-related representations for image peppers at sample density of 3 %: **a** GPRED(4)-CDT, PSNR=33.65; **b** GPRAMA(4)-EC, PSNR=34.23

our focus here is to compare the computational complexity among the different methods. The CPU times for representations of images Lena and peppers with and without GPR are provided in Table 6, where ear clipping mesh patching technique is used in all the GPR-related methods.

It is clear that ED method is the fastest (with low quality) and GPR is the most computationally expensive method (with good quality), and the result is consistent with the existing literatures. For example, at sample density of 3 % for image Lena, ED representation takes only 0.026 s, while GPR method takes 248 s. For image peppers at sample density of 3 %, ED method takes 0.027 s while GPR takes 318 s. The smaller sample density, the longer time GPR needs because more points need to be removed before reaching the desired sample density. For all other methods, the computational cost is lower for smaller sample density.

For representation methods without greedy-point removal technique, $\mathbb{M}_{H,3}$ and $\mathbb{M}_{aniso,3}$ provide better quality (see Table 1) but have higher computational cost than ED method (see Table 6). For example, at sample density of 3 % for image Lena, the $\mathbb{M}_{H,3}$ representation takes 0.45 s and $\mathbb{M}_{aniso,3}$ takes 0.47 s, which are about 16 times more than ED method but only 0.2 % of the time needed by GPR method. Similar results are observed for image peppers. Therefore, $\mathbb{M}_{H,3}$ and $\mathbb{M}_{aniso,3}$ are good balances between ED and GPR. In the meantime, $\mathbb{M}_{aniso,3}$ takes about 0.02 s longer than $\mathbb{M}_{H,3}$ which is due to the extra time needed to compute $\mathbb{M}_{aniso}$ (8) in addition to $\mathbb{M}_{H}$ (5). However, the extra cost is negligible (less than 5 %).

For GPR-related representation methods, both GPRED(5) and GPRAMA(3) provide comparable quality with GPR (see Table 4) but take much less time (see Table 6). For example, at sample density of 3 % for image Lena, GPRED(5) takes 10.5 s and GPRAMA(3) takes 4.93 s. GPRAMA(3) only takes 2 % of the time needed by GPR and less than half of the time needed by GPRED(5). Therefore, GPRAMA method can provide comparable quality with GPRED but with lower computational cost, which makes it another good balance between ED and GPR. The results for image peppers are also similar.

Table 4 Comparison of mesh qualities obtained with various GPR-related methods

Image	Sample density (%)	PSNR (dB)					
		GPR	GPRED(5) -CDT	GPRED(5) -EC	GPR- AMA(2)	GPR- AMA(3)	GPR- AMA(4)
	1.0	29.15	29.33	30.19	29.48	30.48	*30.84*
	2.0	31.81	32.03	32.69	32.13	32.90	*33.23*
Lena	3.0	33.35	33.52	34.12	33.50	34.21	*34.51*
	4.0	34.43	34.60	35.14	34.45	35.11	*35.39*
	1.0	*31.12*	30.22	30.65	29.31	30.53	31.06
	2.0	33.10	32.61	32.86	32.18	32.86	*33.24*
Peppers	3.0	34.01	33.75	33.96	33.41	33.98	*34.23*
	4.0	34.61	34.47	34.73	34.08	34.61	*34.88*
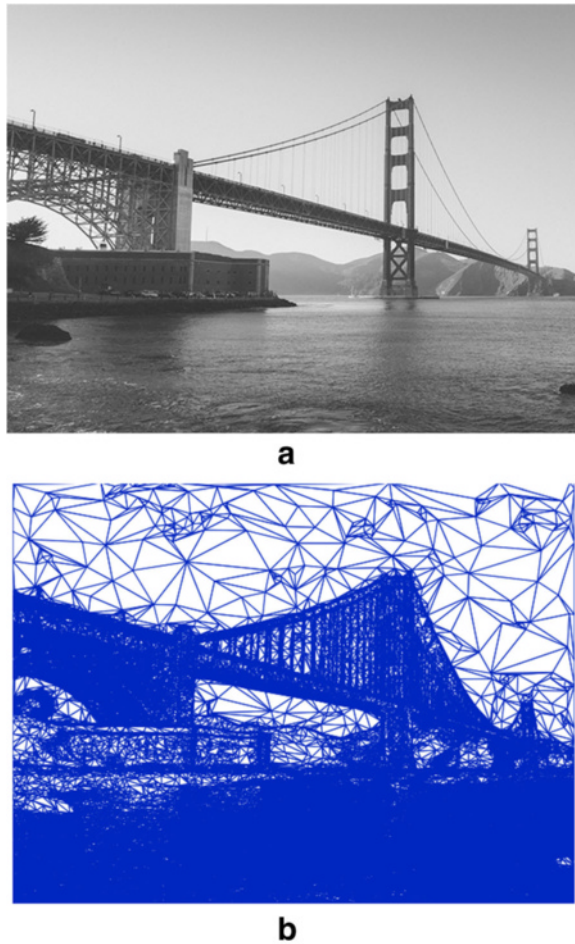
Fig. 14 GPRAMA representation for a image of Golden Gate Bridge at sample density of 1 %: **a** original image, 4000 × 3000; **b** GPRAMA(4), PSNR=34.30

Fig. 15 Reconstructed image of GPRAMA(4) representation for Golden Gate Bridge in Fig. 14a at sample density of 1 %, PSNR=34.30

Table 5 Comparison of mesh qualities for an image of Golden Gate Bridge[a]

Sample density (%)	PSNR (dB)				
	ED	$\mathbb{M}_{aniso,3}$	GPRED(4)	GPRAMA(2)	GPRAMA(4)
0.5	23.84	29.13	31.39	31.67	*32.56*
1.0	27.22	31.03	33.66	33.61	*34.30*

[a]The image pixel resolution is 4000 × 3000

5 Conclusions

Adaptive sampling has become popular in image representation, among which triangular meshes have gained much interest. One common approach is to develop schemes to choose proper sample points then connect the points to form a triangular mesh. Another approach is to generate and adapt the mesh directly to represent the image. In this paper, we have introduced a framework of anisotropic mesh adaptation methods to image representation. The AMA methods take the $\mathbb{M}$-uniform mesh approach and use a metric tensor $\mathbb{M}$ to control the triangular mesh. Firstly, an initial Delaunay triangular mesh is generated based on the desired sample density. Then, the mesh is adapted using the software BAMG according to the provided metric tensor. Lastly, finite element interpolation is used to reconstruct the image from the mesh. The anisotropic metric tensor $\mathbb{M}_{aniso}$ in (8) provides the best representation in this framework among the considered metric tensors. Note that the method proposed by Courchesne et al. [13] (with minor modification) is a special case within this framework.

Within the AMA representation framework, we have developed a GPRAMA method based on the greedy-point removal scheme and a mesh patching technique. The local polygon (may be concave) surrounding a mesh vertex can be triangulated using constrained Delaunay triangulation or ear clipping method. When choosing the initial points using the error-diffusion scheme and CDT is chosen for mesh patching, the corresponding representation method is denoted as GPRED-CDT that is essentially the same as the GPRFS-ED method proposed by Adams [17]; while choosing EC for mesh patching leads to an improved version denoted as GPRED-EC. When starting the initial points from an AMA mesh, in particular, an $\mathbb{M}_{aniso}$ mesh, and EC is chosen for mesh patching, we obtain the GPRAMA representation method that provides better quality than the GPRFS-ED method but with lower computational cost. Overall, mesh patching with EC provides better quality for GPR-related representation methods than mesh patching with CDT. Numerical results on two standard test images, Lena and peppers, are presented, as well as on an image of the Golden Gate Bridge that has higher resolution. The observations are confirmed by the results from three other images. All the

Table 6 Comparison of CPU time (in s) for different representations

Image	Sample density (%)	Without GPR			GPR-related		
		ED	$\mathbb{M}_{H,3}$	$\mathbb{M}_{aniso,3}$	GPR	GPRED(5)	GPR-AMA(3)
	1.0	0.022	0.43	0.40	251	5.23	3.28
	2.0	0.024	0.42	0.44	249	6.97	3.97
	3.0	0.026	0.45	0.47	248	10.5	4.93
Lena	4.0	0.030	0.49	0.51	247	14.2	6.54
	6.0	0.033	0.54	0.57	245	28.9	11.06
	9.0	0.042	0.65	0.69	243	54.6	20.52
	12.0	0.049	0.76	0.80	240	82.8	37.36
	1.0	0.023	0.38	0.40	321	4.89	2.97
	2.0	0.024	0.40	0.42	319	6.73	3.70
	3.0	0.027	0.44	0.47	318	9.92	4.54
Peppers	4.0	0.029	0.47	0.51	317	14.0	6.00
	6.0	0.039	0.53	0.60	315	25.7	10.6
	9.0	0.043	0.63	0.68	313	53.6	15.4
	12.0	0.050	0.72	0.76	309	93.0	34.7

results demonstrate that AMA representation is superior than ED representation, and GPRAMA performs better than the GPRFS-ED method.

AMA representation methods have clear mathematic framework and provides flexibility for both adaptation using different metric tensors and reconstruction using different interpolation methods, although we have only focused on linear finite element interpolation in this paper. The AMA representation of the image and the mesh adaptation strategy will be useful for image scaling and PDE-based image processing such as image smoothing and edge enhancement using anisotropic diffusion filters, which are topics under our current investigation.

Appendix
Finite element interpolation for triangles
This appendix provides a brief introduction to the linear and quadratic finite element interpolation for triangles. The interpolation for quadrilateral elements are similar. Figure 1 shows a triangular element K and an isosceles right triangle $\hat{K}$ as the reference element. The vertices of K are denoted as $\mathbf{a}_1, \mathbf{a}_2$, and $\mathbf{a}_3$, and the midpoints of the corresponding sides are denoted by $\mathbf{a}_4, \mathbf{a}_5$, and $\mathbf{a}_6$. The vertices of the reference element $\hat{K}$ are located at $\hat{\mathbf{a}}_1(0,0), \hat{\mathbf{a}}_2(1,0)$, and $\hat{\mathbf{a}}_3(0,1)$. The midpoints in $\hat{K}$ are located at $\hat{\mathbf{a}}_4(0.5,0), \hat{\mathbf{a}}_5(0.5,0.5)$, and $\hat{\mathbf{a}}_6(0,0.5)$.

Denote the coordinates of the vertices of K as $\mathbf{a}_1(x_1,y_1), \mathbf{a}_2(x_2,y_2)$, and $\mathbf{a}_3(x_3,y_3)$. The corresponding function values are denoted by $f_1 = f(x_1,y_1), f_2 = f(x_2,y_2)$, and $f_3 = f(x_3,y_3)$. For any point $\mathbf{a}(x,y)$ in the element K, the corresponding point $\hat{\mathbf{a}}(\xi,\eta)$ in the reference element $\hat{K}$ is given by

$$\begin{bmatrix} \xi \\ \eta \end{bmatrix} = \begin{bmatrix} x_2 - x_1 & x_3 - x_1 \\ y_2 - y_1 & y_3 - y_1 \end{bmatrix}^{-1} \times \begin{bmatrix} x - x_1 \\ y - y_1 \end{bmatrix}. \quad (14)$$

For linear interpolation, only the values at the three vertices are needed, and the function value at any point (ξ,η) is interpolated as follows

$$f(\xi,\eta) = \sum_{i=1}^{3} f_i \cdot N_i(\xi,\eta), \quad (15)$$

where $N_i(\xi,\eta)$ is the basis functions at $\hat{\mathbf{a}}_i$ and is defined as follows

$$N_1(\xi,\eta) = 1 - \xi - \eta; \quad N_2(\xi,\eta) = \xi; \quad N_3(\xi,\eta) = \eta. \quad (16)$$

For quadratic interpolation, the midpoints of the sides are needed. Denote the function values at midpoints $\mathbf{a}_4, \mathbf{a}_5$, and $\mathbf{a}_6$ as f_4, f_5, and f_6, respectively. Then, the function value at any point (ξ,η) in $\hat{K}$ is interpolated as follows

$$f(\xi,\eta) = \sum_{i=1}^{6} f_i \cdot N_i(\xi,\eta), \quad (17)$$

where the basis functions are defined as follows

$$N_1 = (1 - \xi - \eta)(1 - 2\xi - 2\eta); \quad N_2 = \xi(2\xi - 1);$$
$$N_3 = \eta(2\eta - 1); \quad N_4 = 4\xi(1 - \xi - \eta);$$
$$N_5 = 4\xi\eta; \quad N_6 = 4\eta(1 - \xi - \eta). \quad (18)$$

Authors' contributions
The author has introduced an anisotropic mesh adaptation framework for image representation and developed a GPRAMA method that provides better quality than many existing representation methods. The author has also developed a mesh patching technique for the implementation of the GPRAMA method.

Competing interests
The author declares that he has no competing interests.

References
1. N Dyn, D Levin, S Rippa, Data dependent triangulations for piecewise linear interpolation. IMAJ. Numer. Anal. **10**, 137–154 (1990)
2. D Terzopoulos, M Vasilescu, in *Proceedings of the IEEE Computer Vision and Pattern Recognition Conference (CVPR'91)*. Sampling and reconstruction with adaptive meshes, (1991), pp. 70–75
3. F Davoine, M Antonini, J-M Chassery, M Barlaud, Fractal image compression based on Delaunay triangulation and vector quantization. IEEE Trans. Image Process. **5**(2), 338–346 (1996)
4. H Edelsbrunner, Triangulations and meshes in computational geometry. Acta. Numerica. **9**, 133–213 (2000)
5. G Ramponi, S Carrato, An adaptive irregular sampling algorithm and its application to image coding. Image Vis. Comput. **19**, 451–460 (2001)
6. X Yu, BS Morse, TW Sederberg, Image reconstruction using data-dependent triangulation. IEEE Comput. Graph. Appl. **21**(3), 62–68 (2001)
7. JG Brankov, Y Yang, NP Galatsanos, Image restoration using content-adaptive mesh modeling. in Proc. IEEE Int. Conf. Image Process. **2**, 997–1000 (2003)
8. Y Yang, MN Wernick, JG Brankov, A fast approach for accurate content-adaptive mesh generation. IEEE Trans. Image Process. **12**(8), 866–881 (2003)
9. D Su, P Willis, Image interpolation by pixel-level data-dependent triangulation. Comput. Graph. Forum. **23**(2), 189–201 (2004)
10. JG Brankov, Y Yang, MN Wernick, Tomographic image reconstruction based on a content-adaptive mesh model. IEEE Trans. Med. Imaging. **23**(2), 202–212 (2004)
11. L Demaret, N Dyn, A Iske, Image compression by linear splines over adaptive triangulations. Signal Process. **86 Issue 7**, 1604–1616 (2006)
12. L Demaret, A Iske, Adaptive image approximation by linear splines over locally optimal Delaunay triangulations. IEEE Signal Process. Lett. **13**(5), 281–284 (2006)
13. O Courchesne, F Guibault, J Dompierre, F Cheriet, Adaptive mesh generation of MRI images for 3D reconstruction of human trunk. ICIAR 2007. **LNCS 4633**, 1040–1051 (2007)
14. M Sarkis, K Diepold, in *Proc. IEEE-RAS Int. Conf. Humanoid Robots*. A fast solution to the approximation of 3-D scattered point data from stereo images using triangular meshes, (Pittsburgh PA, 2007), pp. 235–241
15. S Bougleux, Peyré, LD Cohen, in *Proc. ICCV'09*. Image compression with geodesic anisotropic triangulations, (2009), pp. 2343–2348
16. M Sarkis, K Diepold, Content adaptive mesh representation of images using binary space partitions. IEEE Trans. Image Process. **18**(5), 1069–1079 (2009)
17. MD Adams, A flexible content-adaptive mesh-generation strategy for image representation. IEEE Trans. Image Process. **20**(9), 2414–2427 (2011)
18. PJ Frey, F Alauzet, Anisotropic mesh adaptation for CFD computations. Comput. Methods Appl. Mech. Engrg. **194**, 5068–5082 (2005)
19. W Huang, X Li, An anisotropic mesh adaptation method for the finite element solution of variational problems. Fin. Elem. Anal. Des. **46**, 61–73 (2010)
20. X Li, W Huang, An anisotropic mesh adaptation method for the finite element solution of heterogeneous anisotropic diffusion problems. J. Comput. Phys. **229**, 8072–8094 (2010)
21. AV Dam, PA Zegeling, Balanced monitoring of flow phenomena in moving mesh methods. Commun. Comput. Phys. **7**, 138–170 (2010)
22. X Li, W Huang, Maximum principle for the finite element solution of time-dependent anisotropic diffusion problems. Numer. Meth. PDEs. **29**, 1963–1985 (2013)
23. J Wackers, G Deng, A Leroyer, P Queutey, M Visonneau, Adaptive grid refinement for hydrodynamic flows. Comput. Fluids. **55**, 85–100 (2012)
24. W Huang, Metric tensors for anisotropic mesh generation. J. Comput. Phys. **204**, 633–665 (2005)
25. W Huang, Mathematical principles of anisotropic mesh adaptation. Comm. Comput. Phys. **1**, 276–310 (2006)
26. W Huang, RD Russell, *Adaptive moving mesh methods*. (Springer, New York, 2011)
27. F Hecht, Bidimensional anisotropic mesh generator software (2010). https://www.ljll.math.upmc.fr/~hecht/ftp/bamg/bamg-v1.01.tar.gz. Accessed 1 Sept 2015
28. J Weickert, *Anisotropic diffusion in image processing*. (Teubner-Verlag, Stuttgart, Germany, 1998)
29. USC-SIPI, Image database (2010). http://sipi.usc.edu/database. Accessed 8 Sept 2015
30. JR Shewchuk, Delaunay refinement algorithms for triangular mesh generation. Comput. Geom. Theory Appl. **22**(1–3), 21–74 (2002)
31. J O'Rourke, *Computational Geometry in C*. (Cambridge University Press, New York, USA, 1998)