

# Feature-preserving Image Restoration from Adaptive Triangular Meshes

Ke Liu <sup>\*</sup>, Ming Xu, Zeyun Yu

University of Wisconsin Milwaukee, Milwaukee WI 53211, USA

**Abstract.** The triangulation of images has become an active research area in recent years for its compressive representation and ease of image processing and visualization. However, little work has been done on how to faithfully recover image intensities from a triangulated mesh of an image, a process also known as image restoration or decoding from meshes. The existing methods such as linear interpolation, least-square interpolation, or interpolation based on radial basis functions (RBFs) work to some extent, but often yield blurred features (edges, corners, etc.). The main reason for this problem is due to the isotropically-defined Euclidean distance that is taken into consideration in these methods, without considering the anisotropy of feature intensities in an image. Moreover, most existing methods use intensities defined at mesh nodes whose intensities are often ambiguously defined on or near image edges (or feature boundaries). In the current paper, a new method of restoring an image from its triangulation representation is proposed, by utilizing anisotropic radial basis functions (ARBFs). This method considers not only the geometrical (Euclidean) distances but also the local feature orientations (anisotropic intensities). Additionally, this method is based on the intensities of mesh faces instead of mesh nodes and thus provides a more robust restoration. The two strategies together guarantee excellent feature-preserving restoration of an image with arbitrary super-resolutions from its triangulation representation, as demonstrated by various experiments provided in the paper.

**Keywords:** image restoration, image triangulation, radial basis function, anisotropic radial basis function, local structure tensor

## 1 Introduction

Modern imaging technologies often digitize an image into a uniform array of pixels (or voxels in 3D). With uniformly sampling, the sampling density is inevitably too high in regions where intensities change slowly and too low in regions whose intensities change rapidly. Despite the ease of use in both hardware and software developments, uniformly-digitized images often pose challenges in data storage and transmission, as well as image processing, especially in 3D medical images that have been consistently and significantly grown in size in recent years. Evolving from previously commonly-used uniform sampling, non-uniform

---

<sup>\*</sup> Corresponding author. E-mail: keliu@uwm.edu.

sampling and adaptive mesh triangulation of an image has become an active research area in image processing. Image triangulation involves partitioning an image into a collection of non-overlapping small triangles called mesh elements (faces or triangles). This procedure often serves as an image coding method, meaning that an image in pixels is compressed by using a number of “super-pixels”. This method is a compact way to represent images for effective data storage and transmission, and also an efficient way to process and visualize images, especially for 3D images where the number of voxels can be extremely large. In addition, the resulting mesh edges are expected to be well aligned with image featured (edges or corners) in order to maintain a faithful restoration of the original image. Mesh modeling of an image has many applications like image compression [5, 7, 15, 17], motion tracking and compensation [6, 22, 27, 28, 38, 43], image processing by geometric manipulation [19], medical image processing [34], feature detection [14], pattern recognition [29], computer vision [32], restoration [8], tomographic reconstruction [9], interpolation [36, 37], image/video coding [1, 2, 23, 25, 30, 44], video modeling [13] and image retargeting [21].

A common procedure of image triangulation consists of two steps: 1) generating mesh nodes (vertices) by choosing a set of sampling points defined in the image domain, and 2) connecting these mesh nodes by Delaunay triangulation [16]. Delaunay triangulation is a geometric operator and can avoid long and thin triangles that often lead to poor approximations. The selection of sampling nodes, however, is data-dependent, where the connectivity of the triangulation depends on the data set, based on which the mesh nodes are generated. Depending on how to generate mesh nodes, there are two categories of the image triangulation. The first one places mesh nodes inside the image features but near both sides of feature edges. So the triangulated images of this category show double-layer vertices at both sides of feature edges. The second category places mesh nodes directly at the feature edges, thus there are only single-layer vertices defined right on feature edges. Yang et al. [45] employed Floyd-Steinberg error-diffusion (ED) algorithm to place mesh nodes so that their spatial density varies according to the local image content. As a result, the triangulated images fall into category I. Adams [3] employed greedy-point removal (GPR) and error-diffusion scheme together to achieve meshes of quality comparable to the original GPR scheme but at a much lower computational and memory complexities. With the conjunction of smoothing operators, this method produces image triangulation of category I. Adams also proposed a framework in [4] for mesh generation by fixing various degrees of freedom available within that framework. This method performs extremely well and produces meshes of higher quality than the GPR method, and is considered as a method of category I as well. By contrast, Li et al. [26] proposed a modified version of Rippla [31] and Garland-Heckbert (GH) [20] frameworks which can generate single-layer mesh nodes on edges, and this framework generates triangulated images of category II. Another method of this category was proposed by Tu et al. [39], based on constrained Delaunay triangulations. In this method, the approximating function is not re-

quired to be continuous everywhere but with discontinuities being permitted across constrained edges of triangles in triangulation.

Both categories of image triangulation generated by the methods mentioned above have their advantages and disadvantages. For the first category (double-layer vertices), the quality of image restoration is usually better because all vertices are well defined on images features thus the intensities of pixels during image restoration will not be affected by edges. As a result, the edges in the recovered images are sharp and the Peak Signal-to-Noise Ratio (PSNR) is usually larger. While the restoration quality of methods in category I is high enough for subjective quality testing, the two layers must be very close to each other in order to have well-defined and sharp image edges. A consequence of this is that the resulting meshes always contain lots of thin and long triangles between the two layers, which could cause large approximation errors when the meshes are to be used for numerical analysis (like finite element analysis). Additionally, in many applications, the direct communication between different materials should be maintained, meaning that no “cushion” layer between materials should be introduced in the meshes. Moreover, using two layers of mesh vertices usually has more storage cost, resulting more memory space to be used and more time to transmission it over the network. Methods of category II avoid the small triangles and also the “cushion” layer problem, thus the mesh quality is usually better if proper steps are taken. However, the vertices are defined on feature edges, where the nodal intensities are ambiguously defined. That is, the intensity of an edge pixel will be contributed by both sides of the image edge. As will be shown in the experiments, the restored images often suffer from blurred and distorted feature edges if not properly addressed.

Because our proposed method is a single step in image processing, the same mesh will be used for numerical analysis in future. Because the obvious limitation of methods of category I, we are more interested in a method that lies in the second category (single-layer approaches). However, in order to address the blurring and distortion problems often seen in existing approaches in this category, we propose a method based on the radial basis function (RBF) interpolation with the following improvements: 1) rather than considering only the Euclidean distances between vertices, our method also takes into consideration the image local orientations, yielding an anisotropic radial basis function (ARBF). 2) our method does not use intensities of vertices directly, but instead we utilize the intensities of triangles to eliminate the uncertainty of nodal intensities on feature edges.

Our proposed method provides a new approach to restore image from triangular mesh. Because triangular mesh representations of images have much fewer nodes (sampling rates are often as low as 5% – 6%) defined, storage, transmission, and image operations like smoothing, sharpening, etc. will be much faster at the mesh domain instead of pixel domain. Also, our proposed method can be used to visual verification for image operations done in mesh domain.

The remainder of this paper is organized as the following. Section 2.1 briefly summarizes our mesh generation method. Section 2.2 introduces image restora-

tion using traditional RBF interpolation. The proposed image restoration is presented in Section 2.3. Section 2.4 shows the detailed algorithm of our proposed method. Finally, Section 3 shows the experimental results and discussions. Section 4 concludes this paper.

## 2 Methods

While mesh generation from images is not the main focus of the current paper, we will first give a brief summary of this step just for completion of the present work. The traditional (isotropic) radial basis function (RBF) interpolation is then introduced, followed by the proposed anisotropic RBF-based interpolation for image restoration from meshes. The detail of the implementation algorithm is given below as well.

### 2.1 Adaptive Mesh Generation from Images

A series of algorithms are used to generate high quality, feature-sensitive, and adaptive meshes from a given image. Firstly, three kinds of the sample points (namely, Canny’s points, halftoning points, and uniform points) are generated. Secondly, a triangular mesh is generated from these points by using constrained Delaunay triangulation. The Canny’s edge detector is employed to guarantee that important image features are preserved in the meshes. A halftoning-based sampling strategy is adopted to provide feature-sensitive and adaptive point distributions in the image domain. Finally, a Delaunay-triangulation is used to generate initial quality triangulation of the image. These steps are briefly summarized below.

**Canny Sample Points** Image edges are important features in an image and need to be preserved in the obtained meshes. Canny edge detector is a well-known method to deal with boundary extraction. In this paper, we use Canny edge detector to generate the initial Canny edge points and they are strictly attached to the boundary of the features of the image. However, the initial Canny edge points are too dense to yield quality meshes if all these edges are used as mesh nodes. In our method, we take the curvature information of every Canny’s edge point into account and use the Principal Component Analysis (PCA) to determine the sampling density. The PCA method can detect the overall attribute of the neighbors of a certain size by a statistical way. After the PCA sampling, tiny features and features with high curvature have dense sample points and big features or features with straight lines have sparse sampled points.

**Halftoning Sample Points** The edge points generated by the Canny edge detector described above can only capture pixels on or near the image edges. In order to have a decent initial mesh, one has to scatter some more points in the non-edge regions of the image. We adopt the halftoning sample points based on

the approach described in [45]. This method generates the sample points using the second derivatives of an image, where most of the sample points are placed near the image features (edges).

**Uniform Sample Points** Although the halftoning sample points can cover most non-edge regions of the image, it is possible that no point (either Canny or halftoning) is found in regions of almost constant intensities. We therefore generate some points uniformly to cover the rest of the images where the first two types of sample points are not located. A point  $(x, y)$  is said to be a valid uniform sample point if no Canny’s or halftoning points are found in its neighborhood in a fixed distance.

**Mesh Generation via Constrained Delaunay Triangulation** The sample points found above are used to generate our triangular mesh for a given image by using the Delaunay triangulation. We employed a popular open source software *Triangle* [35] for Delaunay triangulation. In order to guarantee the obtained meshes being well aligned with image edge features, we provide to *Triangle* with a set of line segments as additional constraints formed by connecting the Canny’s sample points along the detected Canny’s edges. With all the described strategies combined, we can generate high quality, feature-sensitive, and adaptive meshes from a given grayscale image. Some meshing examples will be shown in the result section below.

## 2.2 Review of Radial Basis Function (RBF) Interpolation

The traditional radial basis function interpolation is given by

$$f(\mathbf{x}) = \sum_{i=1}^N w_i \phi(\|\mathbf{x} - \mathbf{x}_i\|) \quad (1)$$

where the interpolated function  $f(\mathbf{x})$  is represented as a weighted sum of  $N$  radial basis functions  $\phi(\cdot)$ , each centered differently at  $\mathbf{x}_i$  and weighted by  $w_i$ . Let  $f_j = f(\mathbf{x}_j)$ . By given conditions  $f_j = \sum_{i=1}^N w_i \phi(\|\mathbf{x}_j - \mathbf{x}_i\|)$ , the weights  $w_i$  can be solved by

$$\begin{bmatrix} \phi_{11} & \cdots & \phi_{1N} \\ \vdots & \ddots & \vdots \\ \phi_{N1} & \cdots & \phi_{NN} \end{bmatrix} \begin{bmatrix} w_1 \\ \vdots \\ w_N \end{bmatrix} = \begin{bmatrix} f_1 \\ \vdots \\ f_N \end{bmatrix} \quad (2)$$

where  $\phi_{ji} = \phi(\|\mathbf{x}_j - \mathbf{x}_i\|)$ . Once the unknown weights  $w_i$  are solved, the image intensity at an arbitrary pixel can be calculated by

$$s(\mathbf{x}) = \sum_{i=1}^N w_i \phi(\|\mathbf{x} - \mathbf{x}_i\|) \quad (3)$$

In the traditional RBF method, the distance between the point  $\mathbf{x} \in \mathbb{R}^d$  and center  $\mathbf{x}_i \in \mathbb{R}^d$  is measured by Euclidean distance. Let  $r = \|\mathbf{x} - \mathbf{x}_i\|$ , commonly used radial basis functions include:

$$\text{Gaussian} : \phi(r) = e^{-(cr)^2}$$

$$\text{Multiquadric (MQ)} : \phi(r) = \sqrt{r^2 + c^2}$$

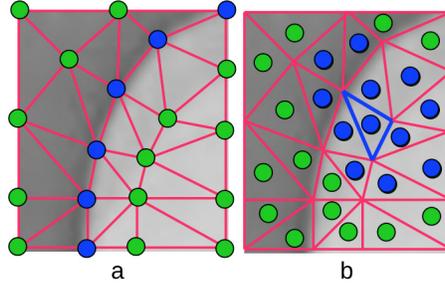
$$\text{Inverse Multiquadric (IMQ)} : \phi(r) = \frac{1}{\sqrt{r^2 + c^2}}$$

$$\text{Thin Plate Spline (TPS)} : \phi(r) = r^2 \ln(r),$$

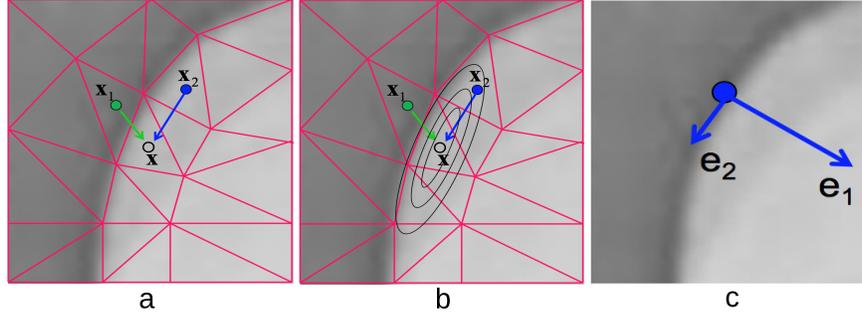
where  $c$  is a shape parameter. The shape parameter plays a major role in improving the accuracy of numerical solutions. In general, the optimal shape parameter depends on the densities, distributions and function values at the nodes. However, it is difficult to assign different shape parameters for each local domain. Thus, choosing shape parameters has been an active topic in approximation theory [42]. Interested readers can refer to [18, 24, 33, 40, 41] for more details.

One question about restoring image from triangular meshes is: what intensities should be used, intensities on vertices or intensities on faces? In the mesh generation approach described above, many vertices are located on image edges. These vertices are good to capture image gradients (or orientations) but not for image intensities because there is an ambiguity in assigning intensity to a node defined on an edge, as illustrated for blue nodes in Fig. 1 (a). Obviously, a very small change (or error) on the location of blues nodes would make a big interpolation difference if the mesh vertices are used as the nodal values in RBFs. A better way is to use face centers as the nodal values for RBF interpolations, which can eliminate the ambiguity and is less sensitive to mesh errors. Fig. 1 (b) shows this idea, where the face centers are more robust to the location changes of mesh vertices. Results of vertex-based RBF interpolation and triangle-based RBF interpolation can be found in Fig. 3 (c) and 3 (d) in Section 3.

Although using face centers performs better than the vertex-based RBF interpolation, the traditional RBF is isotropic in the sense that only the geometrical distance information is considered, which often causes blurring and distortion artifacts as can be seen in Fig. 3 (d). To capture the anisotropy of the image features, the direction of image edges has to be considered as well. Otherwise, nodes across feature edges may have strong influence on the pixel being interpolated. Fig. 2 (a) shows the cause of the blurred edge problem.  $\mathbf{x}$  is the pixel whose intensity we want to find out. The intensities on nodes  $\mathbf{x}_1$  and  $\mathbf{x}_2$  are two of the neighbors used for interpolation. The weights of them are determined only by the Euclidean distance to  $\mathbf{x}$  based on the definition of traditional RBF. However,  $\mathbf{x}_1$  is on the other side of the feature edge, so it should have much less influence on  $\mathbf{x}$  than  $\mathbf{x}_2$ . The isotropic RBF has a hyper-spherical support domain which cannot satisfy this data-dependent requirement. Thus the intensity on  $\mathbf{x}$  is blurred by  $\mathbf{x}_1$ . By contrast, Fig. 2 (b) shows the anisotropic RBF (ARBF) interpolation. The support domain of ARBF is a hyper-ellipsoid. By choosing proper shape parameter, the support domain could rule out the interfering node



**Fig. 1.** Example of interpolation. (a) Interpolation by vertices. Green dots are vertices defined on feature. Blue dots are vertices defined on feature edge. (b) Interpolation by faces. Green dots are face centers. Blue dots are face centers used for interpolation of the intensities of pixels enclosed by the blue triangle.



**Fig. 2.** Interpolation schemes. (a) Isotropic RBF interpolation. (b) Anisotropic RBF interpolation. (c) Eigenvectors on an edge pixel.  $e_1$  shows the normal direction.  $e_2$  shows the tangent direction.

$x_1$ , or give insignificant weight to node  $x_1$ . Thus the blurring effect will be eliminated and sharp features can be well retained. In the following subsections we will elaborate on the detail of designing anisotropic radial basis functions for image restoration.

### 2.3 Anisotropic Radial Basis Function (ARBF) interpolation

The main difference between the isotropic and anisotropic RBFs is the definition of distance metrics used. As in [12], the anisotropic RBF is defined as follows:

**Definition 1.** Given  $N$  distinct points  $X = \{\mathbf{x}_j \in \mathbb{R}^d\}_{j=1,\dots,N}$  and a  $d \times d$  positive definite matrix  $\mathbf{T}$ , the anisotropic radial basis function associated with a radial basis function  $\Phi_j(\cdot) = \phi(\|\cdot - \mathbf{x}_j\|_2)$  is defined by

$$\Phi_{\mathbf{T},j}(\cdot) := \phi(\|\cdot - \mathbf{x}_j\|_{\mathbf{T}}), \quad (4)$$

where  $\|\mathbf{x}\|_{\mathbf{T}} = \mathbf{x}^T \mathbf{T} \mathbf{x}$ .

The support domain of ARBF is hyper-ellipsoid instead of a hyper-sphere in traditional RBF. Its center is  $\mathbf{x}_j$ , associated with the quadratic form  $(\mathbf{x} - \mathbf{x}_j)^T (\mathbf{x} - \mathbf{x}_j)$ . Interested readers can refer to [10, 11] for more details of ARBF.

To construct the metric  $\mathbf{T}$ , we use the image structure tensor

$$G_{\sigma} * \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$

where  $G_{\sigma}$  is the Gaussian smooth operator, and  $\begin{bmatrix} I_x \\ I_y \end{bmatrix}$  is the image gradient at a pixel. Two eigenvectors  $\mathbf{e}_1$  and  $\mathbf{e}_2$  are the normal and tangent directions of the edge, respectively, as shown in Fig. 2 (c). The corresponding eigenvalues are  $\lambda_1$  and  $\lambda_2$ . The anisotropic metric is defined by

$$\mathbf{T} = [\mathbf{e}_1 \mathbf{e}_2] \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} \begin{bmatrix} \mathbf{e}_1^T \\ \mathbf{e}_2^T \end{bmatrix} \quad (5)$$

Similar to the isotropic RBF but with a modified distance metric, the ARBF image interpolation problem becomes

$$s'(\mathbf{x}) = \sum_{i=1}^N w'_i \phi(\|\mathbf{x} - \mathbf{x}_i\|_{\mathbf{T}}) \quad (6)$$

Please note that the matrix in equation (2) should also be updated accordingly with the new distance metric  $\mathbf{T}$ . Therefore, the new set of weights  $w'_i$  would be different from the weights  $w_i$  in the isotropic RBF interpolation.

## 2.4 Algorithms

The following algorithm shows the steps of the proposed approach for image restoration from triangular meshes. The major step is the ARBF interpolation which comprises two sub-steps. First, the weight coefficients are solved by using the new distance metric  $\mathbf{T}$ . As stated in section 2.2, this is done by taking intensities at triangle centers. Then the weights are applied to equation (6) to restore the intensity at each pixel.

Algorithm: Image Reconstruction

```
ImageReconstruction()
{
    // read image mesh
    loadMesh();

    // calculate intensities for each triangle center
    calculateTriangleCenters();
```

```

// find the neighboring vertices and triangles for each node
findNeighbors();

// rescale eigenvalues in equation (5)
calculateEigenvalues();

// calculate metric T in equation (5)
computeMetrics();

// do the ARBF interpolation in equation (6)
ARBFInterpolation();

// output result
printResult();
}

ARBFInterpolation()
{
  for (every triangle centers)
    solveCoefficients();
  end

  for (every triangles)
    for (every pixel in current triangle)
      applyCoefficientsToInterpolation();
    end
  end
}

```

### 3 Results and Discussion

Numerous experiments have been conducted on publicly available images by using the proposed approaches and the image restoration results are all promising. Due to the space limit, we will only consider the well-known “Lena” image and three medical images of different sizes.

Fig. 3 (a) is the original Lena image of size  $256 \times 256$  pixels. Fig. 3 (b) is the result of assigning a constant intensity to all pixels in a mesh triangle (so-called piecewise interpolation). As we can see, this result shows heavy mosaic effect. Fig. 3 (c) is the result of iso-RBF interpolation using intensities on vertices. As previously stated on section 2.3, the ambiguity of intensities on vertices blurred the result. Fig. 3 (d) is the result of iso-RBF interpolation using intensities on triangle centers. In this case, there is no ambiguity of intensities. So the result is much better comparing to Fig. 3 (c). However, the feature edges are still blurred and some distortions are clearly seen because of the lack of directional

information used in isotropic RBF. Fig. 3 (e) is the result of ARBF interpolation using intensities on triangle centers with multi-quadrics (MQ) basis function. The result is much better thanks to a modified distance metric that incorporates both geometric distances and data-dependent feature orientations. Fig. 3 (f) is similar to Fig. 3 (e), except that the basis function is inverse multi-quadrics (IMQ). We have also tested other basis functions like Gaussian and Thin-Plate-Spline (TPS). However, it is hard to find a proper shape parameter to get a reasonable result for Gaussian, and the TPS interpolation doesn't converge.

In Fig. 4, more details of the Lena experiment are shown. (a) is the original Lena image, the same as Fig. 3 (a). (b) is the mesh generated by the method outlined in section 2.1. (c) is the recovered image, which is the same as Fig. 3 (e). To visually see the generated mesh and compare the difference between the original and restored images, Fig. 4 (d)–(f) are the zoomed-in views of (a)–(c), respectively. As the results show, the mesh quality is high enough for subsequent numerical analysis and the the recovered image is very close to the original one. As a matter of fact, the restored image looks smoother due to the smooth radial basis functions used, and the sharp edge features are well preserved. Fig. 5 shows the original brain MRI, its generated mesh, and the result of ARBF interpolation using intensities on triangle centers with the MQ basis function. The zoomed-in views show the quality of mesh and restoration as well. Fig. 6 shows another MRI experiment of breast. Fig. 7 shows a CT-scanning experiment. From all these examples, one can see the effectiveness of the proposed approaches for image mesh generation and feature-preserving restoration.

To give a quantitative evaluation of the restored images, we use the widely-used peak signal-to-noise ratio (PSNR) as defined below:

$$PSNR = 20 * \log_{10} \left( \frac{255}{RMSE} \right), \quad (7)$$

where

$$RMSE = \sqrt{\frac{\sum_{i=0}^{M-1} \sum_{j=0}^{N-1} [O(i, j) - I(i, j)]^2}{M * N}},$$

where  $M$  and  $N$  are the dimensions of the image.  $O(i, j)$  is the original intensity at pixel  $(i, j)$  and  $I(i, j)$  is the interpolated intensity at  $(i, j)$ .

Table 1 gives a summary of the Lena image using different restoration approaches. The compression ratio in the table means the ratio of the number of vertices in the mesh vs. the number of pixels in the original image. As we can see, the restored image with the anisotropic RBF interpolation gives the best PSNR score. Table 2 summarizes the other three data sets, where the running time of image restoration for each case is included and measured on a PC with 1.8 GHz CPU and 2 GB RAM. The proposed algorithms were implemented in C programming and will be released to the public.

## 4 Conclusions

The present paper describes a nonlinear interpolation method by using anisotropic radial basis functions and structure tensor driven metrics. Using the proposed methods, an original image can be stored and processed in the mesh format with some nice advantages including less storage requirement, faster transmission speed, and more efficient image processing due to the significantly reduced number of mesh nodes as opposed to the number of pixels in the original image. The generated meshes, after some post-processing such as mesh-based segmentation, can be readily used for further numerical analysis. The present image restoration algorithm provides an effective way to restore the image with an arbitrary super-resolution from a mesh representation, serving as a decoding algorithm for the mesh-based image coding technique. The anisotropic RBF algorithm can be used as a de-blurring process as well with sharp features well preserved in the images.

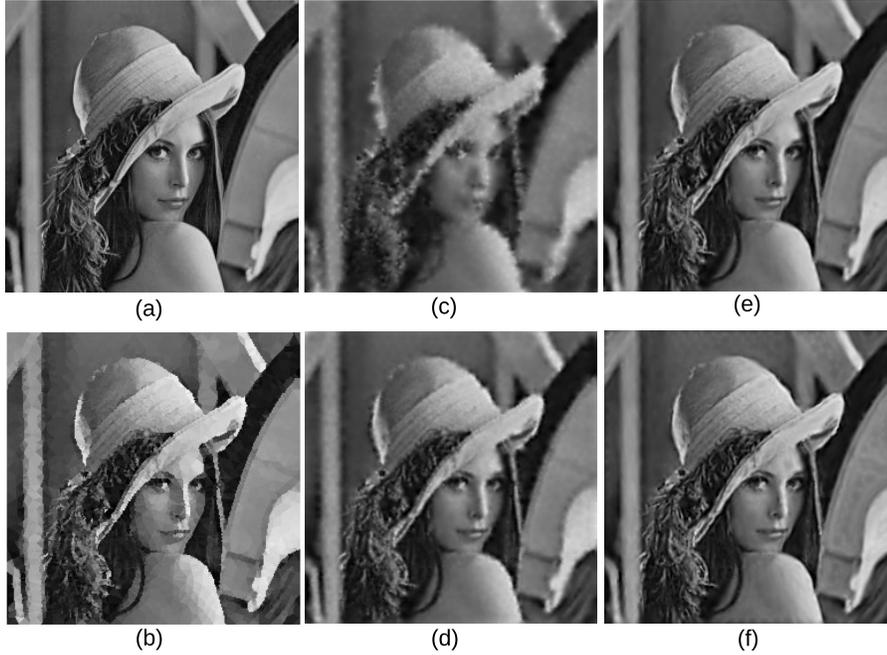
As the image restoration algorithm shows, the time complexity of the function `ARBFInterpolation()` is  $O(m \times n)$ , where  $m$  is the number of triangles and  $n$  is the number of pixels inside of a triangle. In case of 3D images or very large 2D images, the running time could be very expensive. One of our further investigations would be the parallel implementation of the proposed algorithm using GPU programming. Fortunately the present method is very straightforward to parallelize to accelerate the computations. Additionally we are also interested in the mesh-based image segmentation by using the adaptive meshes generated from the original images, and in applying the segmented meshes to image-based numerical analysis.

**Table 1.** Summary of the Lena image (Fig. 3).

Lena (size is $256 \times 256$ , compression ratio is 6%)	PSNR (db)	Shape Parameter
Piecewise Interpolation	22.9703	0.5
Triangle-based ISO-RBF Interpolation	26.7367	0.5
Triangle-based ARBF Interpolation (MQ)	28.2088	0.5
Triangle-based ARBF Interpolation (IMQ)	27.1836	1.8

**Table 2.** Summary of the three medical images (Fig. 5–7).

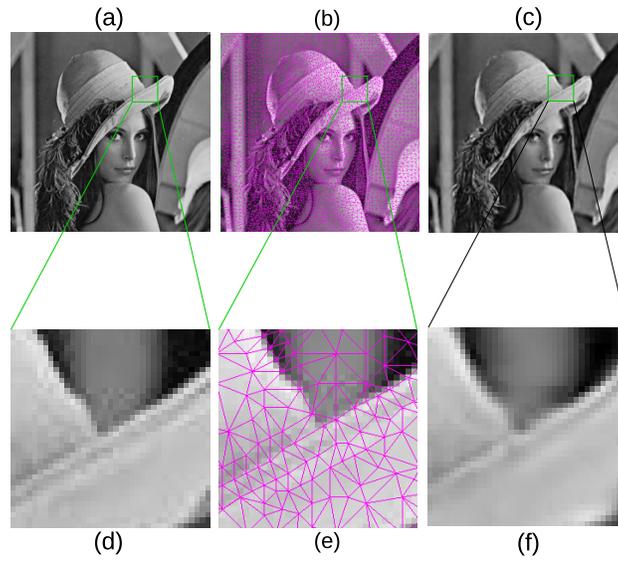
Data	Size	Compression Ratio	PSNR (db)	Shape Parameter	Time (seconds)
Brain	$285 \times 341$	6%	15.7058	0.5	1.71
Breast	$512 \times 512$	5%	11.8763	0.5	8.70
Heart	$356 \times 396$	5%	10.5208	0.5	2.73



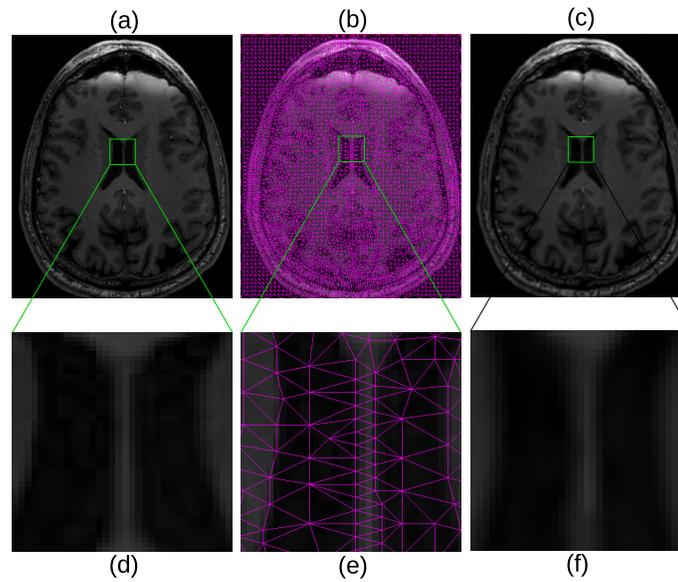
**Fig. 3.** Summary of restoration of Lena. (a) Original Lena image. (b) Result of piecewise interpolation. (c) Result of vertex-based iso-RBF interpolation. (d) Result of triangle-based iso-RBF interpolation. (e) Result of triangle-based ARBF interpolation using MQ basis. (f) Result of triangle-based ARBF interpolation using IMQ basis.

## References

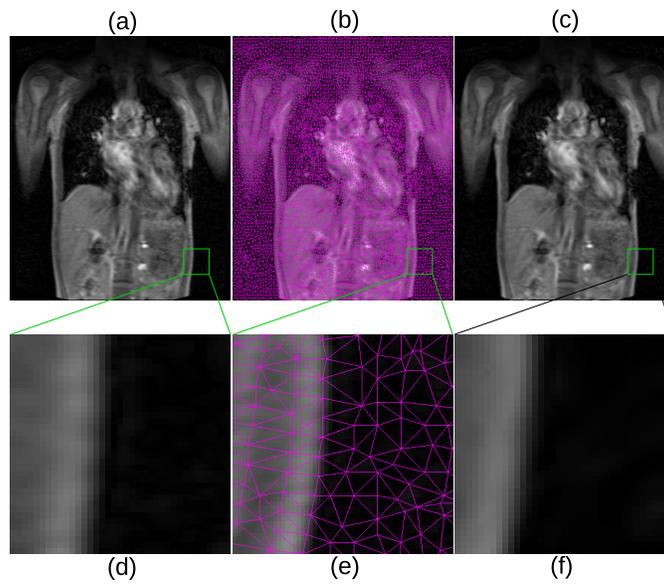
1. Adams, M.: An efficient progressive coding method for arbitrarily-sampled image data. *IEEE Signal Processing Letters* **15** (2008) 629–632
2. Adams, M.: Progressive lossy-to-lossless coding of arbitrarily-sampled image data using the modified scattered data coding method. *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing, Taipei, Taiwan* (2009) 1017–1020
3. Adams, M.: A flexible content-adaptive mesh-generation strategy for image representation. *IEEE Transactions on Image Processing* **20** (2011) 2414–2427
4. Adams, M.: A highly-effective incremental/decremental Delaunay mesh-generation strategy for image representation. *Signal Processing* **93** (2013) 749–764
5. Aizawa, K., Huang, T.: Model-based image coding: advanced video coding techniques for very low bit-rate applications. *Proceedings of the IEEE* **83** (1995) 259–271
6. Altunbasak, Y., Tekalp, A.: Closed-form connectivity-preserving solutions for motion compensation using 2-d meshes. *IEEE Transactions on Image Processing* **6** 533 (1997) 1255–1269



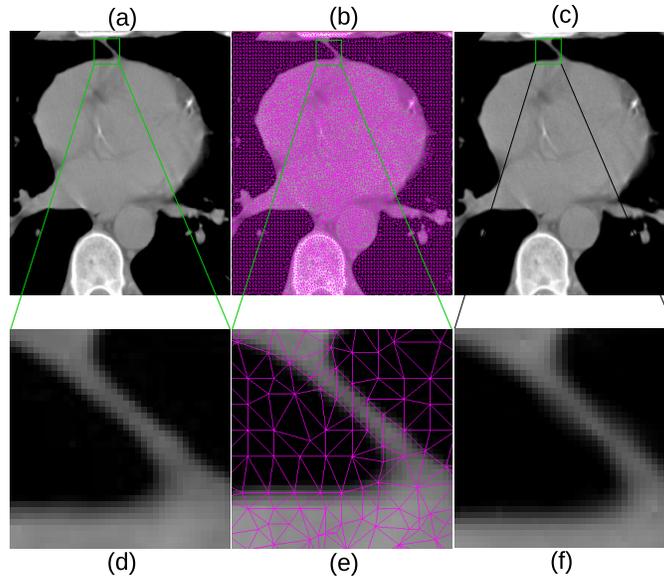
**Fig. 4.** Details of Lena. (a) Original Lena image. (b) Generated mesh of (a). (c) Result of triangle-based ARBF interpolation using MQ basis. (d)–(f) are zoomed-in views of (a)–(c), respectively.



**Fig. 5.** Details of brain MRI. (a) Original brain MRI. (b) Generated mesh of (a). (c) Result of triangle-based ARBF interpolation using MQ basis. (d)–(f) are zoomed-in views of (a)–(c), respectively.



**Fig. 6.** Details of breast MRI. (a) Original breast MRI. (b) Generated mesh of (a). (c) Result of triangle-based ARBF interpolation using MQ basis. (d)–(f) are zoomed-in views of (a)–(c), respectively.



**Fig. 7.** Details of CT-scanned image of heart. (a) Original heart image. (b) Generated mesh of (a). (c) Result of triangle-based ARBF interpolation using MQ basis. (d)–(f) are zoomed-in views of (a)–(c), respectively.

7. Benoit-Cattin, H., Joachimsmann, P., Planat, A., Valette, S., Baskurt, A., Prost, R.: Active mesh texture coding based on warping and DCT. IEEE International Conference on Image Processing, Kobe, Japan (1999)
8. Brankov, J., Yang, Y., Galatsanos, N.: Image restoration using content-adaptive mesh modeling. Proceedings of IEEE International Conference on Image Processing. Volume **2**. (2003) 997–1000
9. Brankov, J., Yang, Y., Wernick, M.: Tomographic image reconstruction based on a content-adaptive mesh model. IEEE Transactions on Medical Imaging **23** (2004) 202–212
10. Casciola, G., Lazzaro, D., Montefusco, L., Morigi, S.: Shape preserving surface reconstruction using locally anisotropic RBF interpolants. Computers & Mathematics with Applications **51** (2006) 1185–1198
11. Casciola, G., Montefusco, L., Morigi, S.: The regularizing properties of anisotropic radial basis functions. Applied Mathematics and Computation **190** (2007) 1050–1062
12. Casciola, G., Montefusco, L., Morigi, S.: Edge-driven image interpolation using adaptive anisotropic radial basis functions. Journal of Mathematical Imaging and Vision **36** (2010) 125–139
13. Chen, J., Paris, S., Wang, J., Matusik, W., Cohen, M., Durand, F.: The video mesh: A data structure for image-based three-dimensional video editing. IEEE International Conference on Computational Photography (ICCP), Pittsburgh, PA, USA (2011) 1–8
14. Coleman, S., Scotney, B., Herron, M.: Image feature detection on content-based meshes. Proceedings of IEEE International Conference on Image Processing. Volume **1**. (2002) 844–847
15. Davoine, F., Antonini, M., Chassery, J., Barlaud, M.: Fractal image compression based on Delaunay triangulation and vector quantization. IEEE Transactions on Image Processing **5** (1996) 338–346
16. Delaunay, B.: Sur la sphere vide. Classe des Science Mathematics et Naturelle **7** (1934) 793–800
17. Demaret, L., Robert, G., Laurent, N., Buisson, A.: Scalable image coder mixing DCT and triangular meshes. IEEE International Conference on Image Processing. Volume **3**, Vancouver, BC, Canada (2000) 849–852
18. Divo, E., Kassab, A.: An efficient localized RBF meshless method for fluid flow and conjugate heat transfer. ASME Journal of Heat Transfer **129** (2007) 124–136
19. Garcia, M., Vintimilla, B.: Acceleration of filtering and enhancement operations through geometric processing of gray-level images. IEEE International Conference on Image Processing. Volume **1**, Vancouver, BC, Canada (2000) 97–100
20. Garland, M., Heckbert, P.: Fast polygonal approximation of terrains and height fields. Technical Report CMU-CS-95-181, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA (1995)
21. Guo, Y., Liu, F., Shi, J., Zhou, Z., Gleicher, M.: Image retargeting using mesh parametrization. IEEE Transactions on Multimedia **11** (2009) 856–867
22. Hsu, P., Liu, K., Chen, T.: A low bit-rate video codec based on two-dimensional mesh motion compensation with adaptive interpolation. IEEE Transactions on Circuits and Systems for Video Technology **11** (2001) 111–117
23. Hung, K., Chang, C.: New irregular sampling coding method for transmitting images progressively. IEEE Proceedings of Vision, Image and Signal Processing **150** (2003) 44–50
24. Kosec, G., Šarler, B.: Local RBF collocation method for Darcy flow. Computer Modeling in Engineering and Sciences **25** (2008) 197–208

25. Lechat, P., Sanson, H., Labelle, L.: Image approximation by minimization of a geometric distance applied to a 3-D finite elements based model. *Proceedings of IEEE International Conference on Image Processing*. Volume **2**. (1997) 724–727
26. Li, P., Adams, M.: A tuned mesh-generation strategy for image representation based on data-dependent triangulation. *IEEE Transactions on Image Processing* **22** (2013) 2004–2018
27. Marquant, G., Pateux, S., Labit, C.: Mesh and “crack lines”: application to object-based motion estimation and higher scalability. *IEEE International Conference on Image Processing*. Volume **2**., Vancouver, BC, Canada (2000) 554–557
28. Nosratinia, A.: New kernels for fast mesh-based motion estimation. *IEEE Transactions on Circuits and Systems for Video Technology* **11** (2001) 40–51
29. Petrou, M., Piroddi, R., Talebpour, A.: Texture recognition from sparsely and irregularly sampled data. *Computer Vision and Image Understanding* **102** (2006) 95–104
30. Ramponi, G., Carrato, S.: An adaptive irregular sampling algorithm and its application to image coding. *Image and Vision Computing* **19** (2001) 451–460
31. Rippa, S.: Adaptive approximation by piecewise linear polynomials on triangulations of subsets of scattered data. *SIAM Journal on Scientific and Statistical Computing* **13** (1992) 1123–1141
32. Sarkis, M., Diepold, K.: A fast solution to the approximation of 3-D scattered point data from stereo images using triangular meshes. *Proceedings of IEEE-RAS International Conference on Humanoid Robots*, Pittsburgh, PA, USA (2007) 235–241
33. Šarler, B., Vertnik, R.: Meshfree explicit local radial basis function collocation method for diffusion problems. *Computers and Mathematics with Applications* **51** (2006) 1269–1282
34. Singh, A., Terzopoulos, D., Goldgof, D.: *Deformable models in medical image analysis*. IEEE Computer Society Press (1998)
35. Shewchuk, J.: Triangle: A two-dimensional quality mesh generator and Delaunay triangulator. <http://www.cs.cmu.edu/quake/triangle.html> (2005)
36. Su, D., Willis, P.: Demosaicing of color images using pixel level data-dependent triangulation. *Proceedings of Theory and Practice of Computer Graphics* (2003) 16–23
37. Su, D., Willis, P.: Image interpolation by pixel-level data-dependent triangulation. *Computer Graphics Forum* **23** (2004) 189–201
38. Toklu, C., Tekalp, A., Erdem, A.: Semi-automatic video object segmentation in the presence of occlusion. *IEEE Transactions on Circuits and Systems for Video Technology* **10** (2000) 624–629
39. Tu, X., Adams, M.: Improved mesh models of images through the explicit representation of discontinuities. *Canadian Journal of Electrical and Computer Engineering* **36** (2013) 78–86
40. Vertnik, R., Šarler, B.: Meshless local radial basis function collocation method for convective-diffusive solid-liquid phase change problems. *International Journal of Numerical Methods for Heat and Fluid Flow* **16** (2006) 617–640
41. Vertnik, R., Šarler, B.: Solution of incompressible turbulent flow by a mesh-free method. *Computer Modeling in Engineering and Sciences* **44** (2009) 65–95
42. Wang, J., Liu, G.: On the optimal shape parameters of radial basis functions used for 2-d meshless methods. *Computer Methods in Applied Mechanics and Engineering* **191** (2002) 2611–2630
43. Wang, Y., Lee, O.: Active mesh – a feature seeking and tracking image sequence representation scheme. *IEEE Transactions on Image Processing* **3** (1994) 610–624

44. Wang, Y., Lee, O., Vetro, A.: Use of 2-D deformable mesh structures for video coding, part II—the analysis problem and a region-based coder employing an active mesh representation. *IEEE Transactions on Circuits and Systems for Video Technology* **6** (1996) 647–659
45. Yang, Y., Miles, N., Jovan, G.: A fast approach for accurate content-adaptive mesh generation. *IEEE Transactions on Image Processing* **12** (2003) 866–881