A Novel Edge-Preserving Mesh-Based Method for Image Scaling

Seyedali Mostafavian and Michael D. Adams
Dept. of Electrical and Computer Engineering, University of Victoria, Victoria, BC V8P 5C2, Canada
alimst@uvic.ca and mdadams@ece.uvic.ca

***Abstract*—In this paper, we present a novel image scaling method that employs a mesh model that explicitly represents discontinuities in the image. Our method effectively addresses the problem of preserving the sharpness of edges, which has always been a challenge, during image enlargement. We use a constrained Delaunay triangulation to generate the model and an approximating function that is continuous everywhere except across the image edges (i.e., discontinuities). The model is then rasterized using a subdivision-based technique. Visual comparisons and quantitative measures show that our method can greatly reduce the blurring artifacts that can arise during image enlargement and produce images that look more pleasant to human observers, compared to the well-known bilinear and bicubic methods.**

I. Introduction

Image scaling, as one of the classical and important problems in digital imaging and computer graphics, has received much attention from researchers. Image scaling refers to the operation of resizing digital images to obtain either a larger or a smaller image, and is required in many applications such as digital photography, computer graphics, and medical imaging. Image scaling is most commonly performed using an image interpolation method. Many different types of interpolation techniques are possible. The effectiveness of an interpolation technique depends on how well it handles undesired effects such as edge blurring and staircasing (which can arise during the scaling process) and how well it preserves the qualitative attributes of the input image.

Image interpolation techniques can be divided into two main categories: raster-based and vector-based. Many different raster-based interpolation approaches have been introduced such as non-adaptive methods like the nearest-neighbor, bilinear, bicubic, and Lanczos methods [1]. They have low complexity and some of them, such as the bicubic method, have become a standard interpolation tool in many image editing programs like Adobe Photoshop. They, however, suffer from severe blurring and artifacts around edges. To address these problems, adaptive interpolation methods have been proposed that consider important image features like edges and textures to produce images with higher visual quality. A survey of raster-based interpolation methods can be found in [1].

Another category of image interpolation techniques is vector-based approaches, in which images are represented with geometric primitives (as opposed to pixels as in raster-based approaches). Using vector-based approaches, image scaling can be performed by simple geometric transformations, with no loss of image quality. One main problem in vector-based interpolation methods, however, is how to create a vector model which faithfully represents the raster image data and its important features such as edges. Among the many techniques to generate a vector image from a raster image, triangle mesh models have become quite popular. With a triangle-mesh model, the image domain is partitioned into a set of non-overlapping triangles called a triangulation. Then, the image intensity function is approximated over each of the triangles. A mesh-generation method is required to choose a good subset of sample points and to collect any critical data from the input image to generate a faithful mesh model.

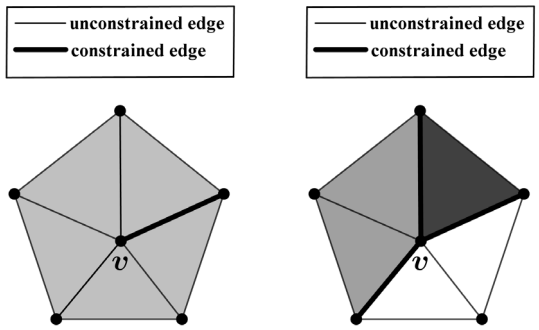
The approximating functions used in mesh models are often piecewise-linear and continuous everywhere (e.g., [2], [3]). Images, however, usually contain many discontinuities (i.e., image edges) and this has encouraged researchers to use mesh models that allow discontinuities in their approximating functions (e.g., [4]–[6]). Such a representation of discontinuities makes the mesh model a good candidate for image scaling because the edges can be reconstructed with the least amount of blurring regardless of how large the scaling factor is. In this paper, we propose a new mesh-based image scaling method. This method first produces a mesh model of the image to be scaled, known as an ERD model [4], [5], which explicitly represents image discontinuities. The model is then rasterized using a subdivision-based technique to obtain the enlarged image. Through experiments, we show that the scaled images obtained with our proposed method are of higher subjective quality (e.g., have much sharper edges) than those obtained with the commonly-used bilinear and bicubic methods.

The remainder of this paper is organized as follows. Section II presents the proposed scaling method, introducing the ERD mesh model and explaining mesh generation and subdivision-based image reconstruction. Next, Section III evaluates the performance of our method by visual comparisons and quantitative measures against the bilinear and bicubic methods. Finally, Section IV concludes with a summary of our work.

II. Proposed Image Scaling Method

Before we can introduce our proposed image scaling method, we must first introduce the type of mesh model that it employs. In what follows, the cardinality of a set S is denoted by $|S|$. The constrained Delaunay triangulation [7] of a set P of points and set E of edge constraints is denoted

978-1-5386-0700-8/17/$31.00 ©2017 IEEE



(a) (b)

Fig. 1: The relationship between vertices, constrained edges, and wedges. Each wedge is colored with a different shade of gray. The (a) single-wedge and (b) multiple-wedge cases.

by $cdt(P, E)$. Consider an image function ϕ defined on the rectangular region $\Gamma = [0, W-1] \times [0, H-1]$. An ERD triangle mesh model of ϕ consists of: 1) a set $P = \{p_i\} \subset \Gamma$ of sample points, 2) a set E of edge constraints, and 3) a set Z of what are called wedge values (to be explained later). As a matter of terminology, the size of the model is defined as $|P|$. The ERD model is associated with the triangulation $cdt(P, E)$ and with an approximating function defined over each face in this triangulation. By combining the functions over all faces, a function $\hat{\phi}$ is obtained that approximates ϕ over the entire image domain Γ. In the ERD model, a set of consecutive faces in a loop around a vertex $v \in P$ that are not separated by any constrained edge is called a *wedge*. Fig. 1 illustrates this definition and the relationship between wedges, vertices, faces, and constrained edges. Each wedge has associated with it what is called a *wedge value*. The wedge value z of the wedge w belonging to vertex v specifies the limit of $\hat{\phi}(p)$ as p approaches v from points inside the wedge w. The approximating function defined over each face f in the ERD model is a linear function that interpolates the three wedge values of f corresponding to its three vertices. With the help of the wedge values, $\hat{\phi}$ in the ERD model can represent discontinuities along constrained edges, which are used to represent the image edges.

Having introduced the ERD mesh model, we now present our proposed image scaling method. The proposed method is comprised of two stages (in order): 1) mesh generation, which produces an ERD mesh model of the image to be scaled; and 2) image reconstruction, which produces a scaled image from the mesh model. In what follows, each stage will be described in detail.

A. Mesh Generation

As mentioned earlier, in the first stage of our image scaling method, we generate a mesh model of the image to be scaled. In what follows, we explain how this mesh-generation process works. The mesh generator in our work is based on earlier work from [8] and [5], but with critical improvements such the addition of a more effective edge detector, a backfilling technique, and subdivision. In mesh generation, we select the parameters of the ERD model (i.e., P, E, Z) to obtain the best possible approximation $\hat{\phi}$ of the input image ϕ for a specific target number N of sample points. The input image to the mesh-generation algorithm is an image function ϕ that is known only at the points in $\Lambda = \{0, 1, ..., W-1\} \times \{0, 1, ..., H-1\}$ (i.e., a truncated integer lattice of width W and height H) and the outputs are the ERD model parameters $P \subset \Lambda \subset \Gamma$, E, and Z. The mesh-generation process in our work consists of the following steps (in order):

1) *Initialization.* Set the iteration number i to 0.
2) *Initial triangulation selection.* Select the values for the initial set P_0 of sample points and E (to be explained later). Then, the initial mesh associated with the triangulation $cdt(P_0, E)$ is constructed.
3) *Wedge-value initialization.* Compute the initial wedge values (to be explained later).
4) *Point selection.* Select a new sample point $q \in \Lambda$ to add to the mesh (to be explained later).
5) *Point insertion.* Insert q in the triangulation. Calculate the wedge values of the affected wedges in the mesh (to be explained later) and select the set P_{i+1} of sample points for the next iteration as $P_i \cup \{q\}$.
6) *Stopping criterion.* If $|P_{i+1}| < N$, set i to $i+1$ and go to step 4. Otherwise, set the output parameter $P = P_{i+1}$ and stop.

Selection of P_0 and E. In the step 2 of mesh generation, P_0 and E are selected as follows. Apply the Canny edge detector [9] with low and high thresholds t_l and t_h, respectively, to the input image ϕ defined on Λ to produce a binary image B, where t_h is computed using the Otsu thresholding method [10] and $t_l = \frac{1}{2} t_h$. Then, from B, a collection of polylines is generated to represent edges. The polylines are then approximated using the simplification method from [11], with the tolerance $\varepsilon = 0.5$, obtaining the set D of simplified polylines. The set P_0 is then chosen as the union of all vertices in D, plus the extreme convex-hull points of Λ (i.e., the four corner points of the bounding box of ϕ). Then, E is selected as the union of all the line-segments in D.

Wedge-Value Calculation. As mentioned earlier, wedge values are used to define the approximating function $\hat{\phi}$ over each triangle. So, the wedge values chosen in steps 3 and 5 above are critically important and are computed as follows. If the vertex $v \in P$ has zero or one incident constrained edge (e.g., as in Fig. 1(a)), its single wedge value z is simply chosen as the value of ϕ at v (i.e., $z = \phi(v)$). Otherwise, if v has more than one incident constrained edge (as shown in Fig. 1(b)), the wedge value associated with each wedge is calculated based on a backfilling technique as follows. Consider a wedge w incident to a vertex v, as shown in Fig. 2(a), and a set S of vertices that are connected to v by unconstrained edges from inside w and are not incident on any constrained edges in the mesh. The wedge value z associated with w is calculated as

$$z = \frac{1}{|S|} \sum_{p \in S} \phi(p), \tag{1}$$

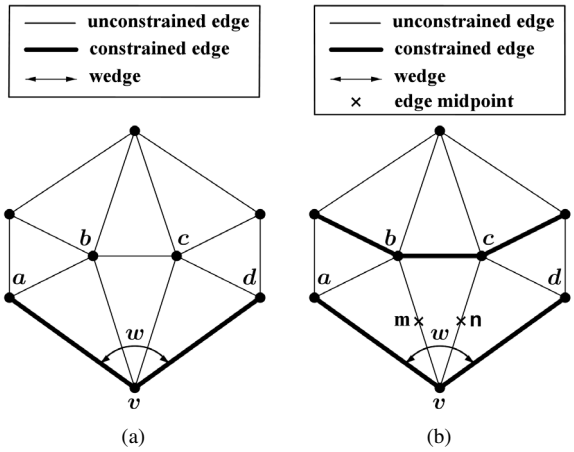


Fig. 2: Two examples of the set S for calculating the wedge value associated with the wedge w. The case (a) where $S = \{b, c\}$ and case (b) where S is empty.

if $|S| \neq 0$. In the special case that $|S| = 0$ (i.e., S is empty), z is calculated as the average of ϕ at the midpoints of the unconstrained edges incident to v from inside w. For the example shown in Fig. 2(a), $S = \{b, c\}$ and z is calculated using (1). In Fig. 2(b), however, S is empty and $z = \frac{\phi(m)+\phi(n)}{2}$, where m and n are the midpoints of the edges $\overline{vb}$ and $\overline{vc}$, respectively. If midpoints (e.g., m and n in Fig. 2(b)) are not in Λ, the bilinear interpolation technique [1] is used to find the corresponding values of ϕ at the midpoints. The set S in (1) is chosen in such a way that it does not include the vertices that correspond to any points that fall on, or are too close to, the image edges detected in step 1 of the mesh-generation process. This is because most often there is some blurring near the image edges and if S includes the vertices corresponding to the pixels in the vicinity of edges, this blurring effect will become part of the mesh model. Then, during the image scaling, this blurring effect will be magnified, leading to undesired blurring and distortions around the reconstructed edges. Thus, to reduce the edge blurring, in our method, we avoid selecting sample points that correspond to the edge pixels when selecting new points for insertion later in step 4.

Selection of q. In step 4 of mesh generation, the point q is chosen in two steps. First, the face f^* into which to insert a point is selected as given by

$$f^* = \underset{f \in F}{\operatorname{argmax}} \sum_{p \in \Omega_f} \left(\hat{\phi}(p) - \phi(p)\right)^2, \tag{2}$$

where Ω_f is the set of all valid points in Λ belonging to face f and F is the set of all faces for which $|\Omega_f| \neq 0$. A point in Ω_f is valid if it does not correspond to any 8-connected pixels of any image edges detected in step 1. Next, a point q in f^* for insertion in the mesh is selected as given by

$$q = \underset{p \in \Omega_{f^*}}{\operatorname{argmax}} \left|\hat{\phi}(p) - \phi(p)\right| \tag{3}$$

(i.e., q is the valid point in f^* with the greatest absolute error).

B. Image Reconstruction

Having introduced the mesh-generation process, we now describe the second stage of our image scaling method, namely, image reconstruction. The image reconstruction process consists of two steps (in order): 1) mesh refinement and 2) mesh rasterization. In step 1, the ERD mesh model obtained from the mesh-generation stage (in Section II-A) is refined iteratively through a process called subdivision. In step 2, the refined mesh model is rasterized to produce the scaled version of the input image. Although, in principle, we could simply rasterize the mesh obtained from the mesh-generation process (i.e., omit the mesh-refinement step), this would lead to an image reconstruction with undesirable artifacts. In particular, the image discontinuity contours in the scaled image would lack a desirable level of smoothness as well as the image function itself. By employing subdivision-based mesh refinement, however, we are able to obtain a mesh with better smoothness. Now, in what follows, the details of each step in the image reconstruction process will be elaborated.

Mesh Refinement. In the first step of the image reconstruction process, the mesh is refined using a variation of the Loop subdivision scheme proposed in [6]. This variation of Loop subdivision was chosen as it is able to selectively smooth only certain parts of the mesh, allowing the discontinuities in our ERD mesh model to be maintained (i.e., not smoothed). In each iteration of subdivision, each triangle face is split into four sub-triangles. Then, the vertex positions and wedge values in the refined mesh are computed as weighted averages of the nearby vertex positions and wedge values in the unrefined mesh as described in [6]. The subdivision process is repeated l times iteratively, resulting in a refined mesh. In order to apply the subdivision scheme, each vertex must be labeled as one of *smooth*, *tear*, or *corner*, and each edge must be labeled as one of *smooth* or *tear*. To use this subdivision scheme with our ERD model, a correspondence between the vertices/edges in our mesh model and the preceding vertex/edge labels in the subdivision scheme is established as follows: 1) a vertex with no incident constrained edges is deemed to be a smooth vertex; 2) a vertex with exactly two incident constrained edges is treated as a tear vertex; 3) a vertex with either one or three or more incident constrained edges is deemed to be a corner vertex; and 4) unconstrained and constrained edges are deemed to be smooth and tear edges, respectively. With the preceding correspondence between the vertices and edges in our mesh model and the labels in the subdivision scheme having been established, the subdivision scheme from [6] can be applied to our model. In our method, subdivision is applied three times (i.e., $l = 3$) to produce a sufficiently-smooth surface.

Mesh Rasterization. In the second step of the image reconstruction process, the subdivided mesh model obtained in step 1 (i.e., the mesh-refinement step) is rasterized, using the piecewise-linear interpolating function associated with the model as described in Section II, to produce the scaled version of the input image. Since the ERD model explicitly represents discontinuities, edges in the reconstructed image

TABLE I: Test images

Image	Size, Bits/Sample
backlight	1400×1400, 8
bowling	1200×1400, 8
frangipani	1920×1280, 8

could generate aliasing effects that are undesirable to the human visual system. Thus, a 3×3 uniform super-sampling technique similar to the one used in [5] is employed during rasterization to avoid such aliasing effects at the edges.

III. Results

Having introduced our proposed image scaling method, we now evaluate its performance by comparing it with the commonly-used bilinear and bicubic interpolation methods. For test data, in this paper, we focus our attention on the representative set of three images listed in Table I, which were deliberately chosen to include images with low, medium, and high levels of detail.

Since numerous approaches could potentially be used to evaluate the performance of image scaling methods, we need to explain the particular approach used herein. To compare image scaling methods for a particular test image I and an image scaling factor α, we proceed as follows. From the high-resolution image I, we first generate a lower-resolution version L of the image, where the resolution has been reduced by a factor of α. This reduction in resolution is performed using the default scaling approach called "pixel mixing" provided by the Netpbm toolkit [12]. Then, each of the image scaling methods under comparison (i.e., the bilinear, bicubic, and proposed methods) is used to increase the resolution of the input image L by a factor of α to produce a new high-resolution image I'. The resulting (upscaled) image I' obtained for each method is then compared to I, which can be viewed as the ground truth. This comparison is performed through both a visual inspection and the percentage edge error (PEE) metric from [13].

In our work, the PEE metric is used because it reflects the quality and sharpness of the edges in the reconstructed image more effectively than other popular and widely-used measures such as the peak signal-to-noise ratio (PSNR). It is a well-established fact that the PSNR measure does not necessarily reflect the human visual perception of image quality [14]. The PEE metric, however, measures how close the edges of the scaled image are to the edges of the high-resolution (ground-truth) image. Generally, the larger positive PEE values indicate that the edges in the scaled image are over-smoothed and more blurring has happened during the scaling process. Therefore, a decrease in the PEE value is a strong indication of a reduction in the blurring effect.

Using the methodology described above, we generated results for each of the three images listed in Table I using each of the bilinear, bicubic, and proposed methods. The results for each test image are illustrated in Figs. 3 to 5 and the corresponding PEE values are given in each figure caption. In the case of the proposed method, results were obtained with the sampling density of 2% for the backlight and bowling images (in Figs. 3 and 4) and 4% for the frangipani image (in Fig. 5). Each of the figures from Figs. 3 to 5 contains the high-resolution image along with a magnified region of interest from it and the same region in the (scaled) images obtained as output from each of the bilinear, bicubic, and proposed methods. Also, for the benefit of the reader, each of the above figures includes the subdivided mesh corresponding to the magnified region obtained from the proposed method.

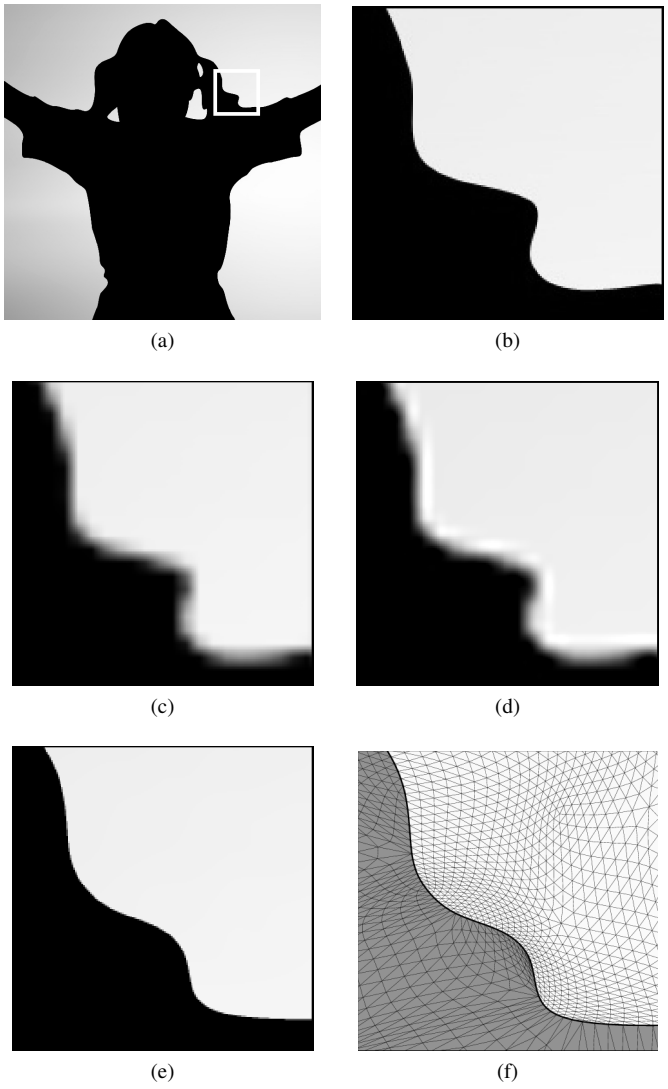


Fig. 3: Scaling results obtained by the bilinear, bicubic, and proposed methods for backlight image with $\alpha = 8$. (a) The high-resolution image with region of interest marked with white rectangle. The same region obtained from the (b) high-resolution image, (c) bilinear (PEE 55.17), (d) bicubic (PEE 47.58), and (e) proposed methods (PEE 0.95). (f) The subdivided mesh corresponding to the result shown in (e), with the constrained edges shown as thick lines.

In Fig. 3, by comparing the magnified region of interest in the high-resolution image in Fig. 3(b) with the corresponding

region in the scaled images obtained from the bilinear (in Fig. 3(c)), bicubic (in Fig. 3(d)), and proposed (in Fig. 3(e)) methods, it is obvious that edges produced with the bilinear and bicubic methods in Figs. 3(c) and (d) are severely blurred and distorted. The result obtained from the proposed method (in Fig. 3(e)), however, shows a much more accurate and sharper reconstruction of edges. Furthermore, the significant drop in the PEE of the results obtained from the proposed method (i.e, the PEE of 0.95) compared with the PEE obtained from the bilinear and bicubic methods (i.e., the PEE of 55.17 and 47.58) correlates well with the improved visual quality observed in the case of our proposed method in Fig. 3(e).

In Fig. 4, given the magnified region of the high-resolution image in Fig. 4(b) as the ground-truth image, we can see that the result obtained from the proposed method in Fig. 4(e) shows much fewer blurring artifacts, specifically along the contours of the arm and ball, compared to the results obtained from the bilinear and bicubic methods in Figs. 4(c) and (d), respectively. Also, the PEE of 0.18 for the result obtained from the proposed method implies better subjective quality compared to the PEE of 27.51 and 18.44 for the results obtained from the bilinear and bicubic methods, respectively.

Similarly, in Fig. 5, given the ground-truth image in Fig. 5(b), we observe that the result obtained from the proposed method in Fig. 5(e) is of higher subjective quality (e.g., much sharper reproduction of the image edges) compared to the results obtained from the bilinear and bicubic methods shown in Figs. 5(c) and (d), respectively. Similar to the results for previous test cases, the lowest PEE is achieved with the proposed scaling method. The PEE of -0.42 for the result obtained from the proposed method is an implication of the sharper and more accurate edge reproduction compared with the PEE of 19.25 and 11.22 for the results obtained from the bilinear and bicubic methods, respectively. The negative sign for the PEE value shows that the reconstructed (scaled) image is sharper than the ground-truth image and it does not necessary indicate a weak edge reproduction, as long as the general structure of the image is maintained well and the absolute value of the PEE is close enough to zero.

Based on the results for the three test cases above, we conclude that the proposed method significantly outperforms the bilinear and bicubic methods in terms of both subjective and objective measures. The better performance in the case of the proposed method is firstly because of the fact that a mesh-based image representation is used which is independent of the image resolution and can be upscaled by simply applying geometric transformations to the mesh primitives. Secondly, and more importantly, we use the ERD mesh model which, by allowing discontinuities in the approximating function, is capable of reconstructing as sharp edges as possible. Furthermore, through the backfilling-based mesh-generation method, the undesired effects of the pixels falling in the vicinity of edges are minimized in our model. Finally, through the subdivision-based image reconstruction, we ensure that the reconstructed surface and edge contours look smooth enough in the scaled image.

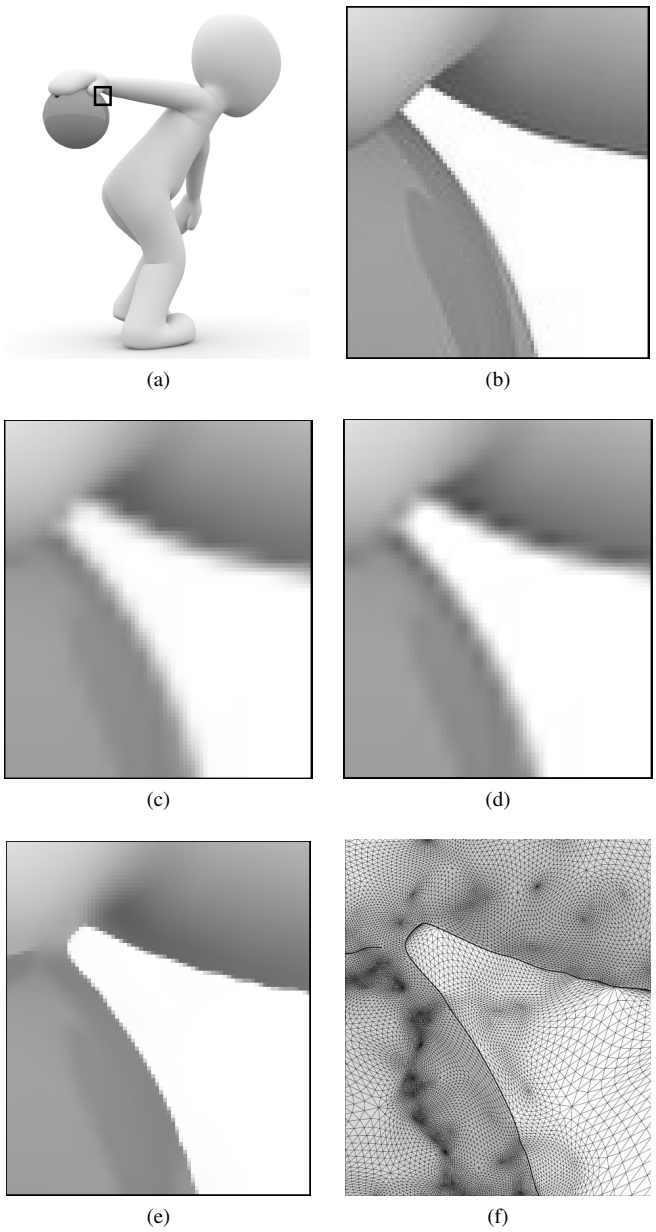


Fig. 4: Scaling results obtained by the bilinear, bicubic, and proposed methods for bowling image with $\alpha = 4$. (a) The high-resolution image with region of interest marked with white rectangle. The same region obtained from the (b) high-resolution image, (c) bilinear (PEE 27.51), (d) bicubic (PEE 18.44), and (e) proposed methods (PEE 0.18). (f) The subdivided mesh corresponding to the result shown in (e), with the constrained edges shown as thick lines.

IV. Conclusions

In this paper, we proposed a new edge-preserving mesh-based method for image scaling. Our proposed method employs a mesh-generation process to create a mesh model which explicitly represents discontinuities in the image. Then, using a subdivision-based image reconstruction approach, the model

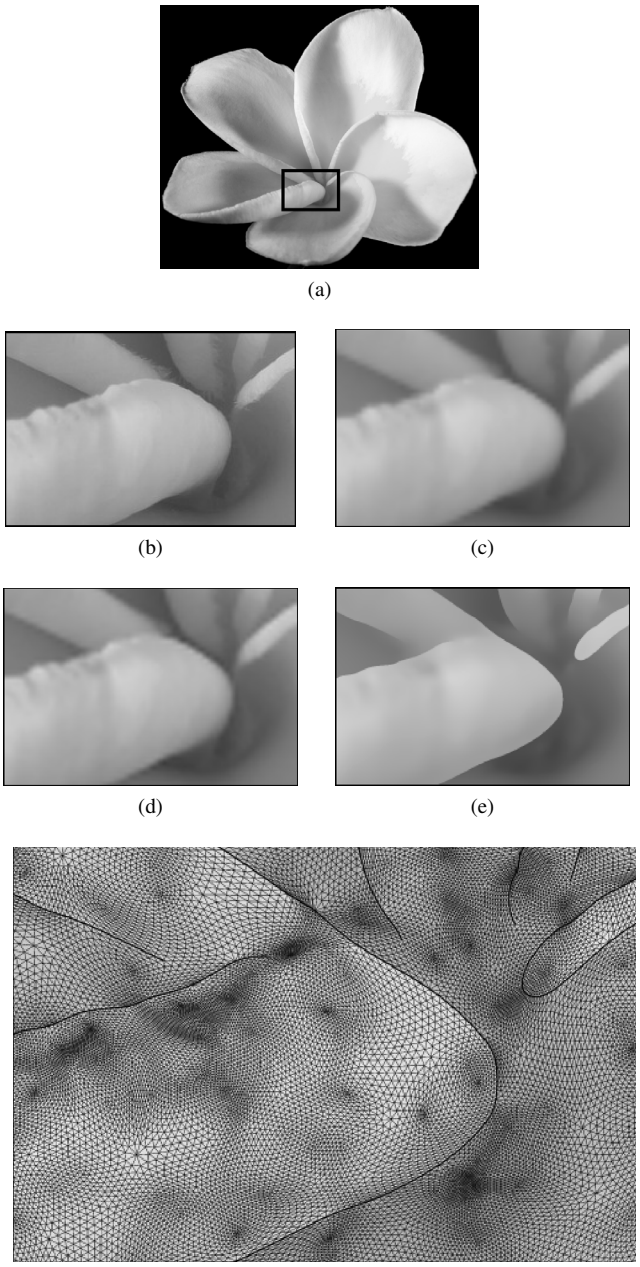


Fig. 5: Scaling results obtained by the bilinear, bicubic, and proposed methods for frangipani image with $\alpha = 4$. (a) The high-resolution image with region of interest marked with white rectangle. The same region obtained from the (b) high-resolution image, (c) bilinear (PEE 19.25), (d) bicubic (PEE 11.22), and (e) proposed methods (PEE -0.42). (f) The subdivided mesh corresponding to the result shown in (e), with the constrained edges shown as thick lines.

is rasterized with a piecewise-linear approximating function to produce the scaled image. We compared the performance of the proposed method against the well-known bilinear and bicubic methods. The evaluation results showed that our proposed method performs more effectively by producing scaled images of higher subjective quality (e.g., more accurate and sharper edges with a great reduction in the blurring artifacts) compared to the results obtained with the bilinear and bicubic methods. Visual comparisons and quantitative measures both demonstrate the better performance of the proposed method. Thus, our new image scaling method can benefit the many image processing applications where image scaling is required such as digital photography, computer graphics, and medical imaging.

REFERENCES

[1] V. Patel and K. Mistree, "A review on different image interpolation techniques for image enhancement," *International Journal of Emerging Technology and Advanced Engineering*, vol. 3, pp. 129–133, Dec. 2013.

[2] H. Zhou, J. Zheng, and L. Wei, "Representing images using curvilinear feature driven subdivision surfaces," *IEEE Transactions on Image Processing*, vol. 23, no. 8, pp. 3268–3280, Aug. 2014.

[3] R. Verma, R. Mahrishi, G. K. Srivastava, and S. Rajesh, "A fast image reconstruction algorithm using significant sample point selection and linear bivariate splines," in *TENCON 2009 - 2009 IEEE Region 10 Conference*, Jan. 2009, pp. 1–6.

[4] S. Mostafavian and M. D. Adams, "An optimization-based mesh-generation method for image representation," in *2015 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing*, Aug. 2015, pp. 234–239.

[5] X. Tu and M. Adams, "Improved mesh models of images through the explicit representation of discontinuities," *Canadian Journal of Electrical and Computer Engineering*, vol. 36, no. 2, pp. 78–86, Spring 2013.

[6] Z. Liao, H. Hoppe, D. Forsyth, and Y. Yu, "A subdivision-based representation for vector image editing," *IEEE Transactions on Visualization and Computer Graphics*, vol. 18, no. 11, pp. 1858–1867, Nov. 2012.

[7] L. P. Chew, "Constrained Delaunay triangulations," *Algorithmica*, vol. 4, pp. 97–108, 1989.

[8] M. Adams, "An evaluation of several mesh-generation methods using a simple mesh-based image coder," in *15th IEEE International Conference on Image Processing*, Oct. 2008, pp. 1041–1044.

[9] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, no. 6, pp. 679–698, Nov. 1986.

[10] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62–66, Jan. 1979.

[11] D. H. Douglas and T. K. Peucker, "Algorithms for the reduction of the number of points required to represent a digitized line or its caricature," *Cartographica: The International Journal for Geographic Information and Geovisualization*, vol. 10, no. 2, pp. 112–122, 1973.

[12] Netpbm homepage: http://netpbm.sourceforge.net/, 2017.

[13] A. S. Al-Fohoum and A. M. Reza, "Combined edge crispiness and statistical differencing for deblocking JPEG compressed images," *IEEE Transactions on Image Processing*, vol. 10, no. 9, pp. 1288–1298, Sep. 2001.

[14] Z. Wang and A. C. Bovik, "Mean squared error: Love it or leave it? a new look at signal fidelity measures," *IEEE Signal Processing Magazine*, vol. 26, no. 1, pp. 98–117, Jan. 2009.