A Progressive Refinement Approach for Image Magnification

Henry Johan Tomoyuki Nishita

The University of Tokyo
{henry, nis}@is.s.u-tokyo.ac.jp

Abstract

The rapid growth in computer graphics and digital cameras has resulted in computer users being able to easily produce digital images. As a result, the need to display and print digital images has increased. Nowadays, high-resolution display and printing devices are available to users. Therefore, high-resolution images are needed in order to produce high quality displayed images and high quality prints. However, since high-resolution images are not usually provided, there is a need to magnify the original images. Previous methods on magnifying images have the disadvantage that either the sharpness of the edges cannot be preserved or that some distinct artifacts are produced in the magnified image. In this paper, we present a novel method for doubling the size of images in which the sharpness of the edges is preserved without introducing distinct artifacts in the magnified images. The proposed method consists of two steps, first generation of an initial magnified image and then progressively refining this image to produce a high quality magnified image. The experimental results show that with the proposed method it is possible to produce magnified images of comparable, and in some cases superior, visual quality to those produced using previous methods.

1. Introduction

Nowadays, there is a large amount of digital images available to computer users. This is caused by the rapid growth both in computer hardware and software technologies. Low price digital cameras are now widespread, and as a result users are able to buy them and take as many digital images as desired. The significant development in the field of computer graphics has also boosted the production of digital images.

As computer users become more familiar with digital images, the need to display and print them also increases. In an era where high-resolution display and printing devices are common, it is crucial that high-resolution images are available in order to produce high quality displayed images and high quality prints. This is particularly important for desktop publishing, large artistic printing, etc. The problem is that high-resolution images are not usually provided. In these cases, there is a need to magnify the original images. Therefore, the development of a good image magnification algorithm is very important.

Until now, a large number of methods for magnifying images have been proposed. However, previous work on magnifying images has the disadvantage that either the sharpness of the edges cannot be preserved or that some highly visible artifacts are produced in the magnified image. This paper presents a method for magnifying images that produces high quality images in the sense that the sharpness of the edges are preserved without introducing distinct artifacts in the magnified images.

The main contribution of this paper is a novel progressive approach for doubling the size of an input image. Our approach starts by creating an initial magnified image taking into account the derivatives between the pixel values during interpolation (considering edges in an implicit way). The initial magnified image is then progressively refined to increase the sharpness of the magnified image.

2. Related work

The simplest way to magnify images is by using the pixel replication method. However, the resulting magnified images have jagged edges. More elaborate approaches use the bilinear or the bicubic interpolation. Commercial software Adobe Photoshop [1] provides these two functions for interpolating images. Other methods, using the B-spline interpolators [8, 22, 11, 21] or the cubic convolution methods [10, 23, 24] have also been proposed. However, these methods tend to blur the edges and cause them to be jagged.

Recently, research on interpolating images taking into account the edges has gained much attention. Allebach and Wong [3] and Salisbury *et al.* [19] proposed methods that search for edges in the input image and use them to assure that the interpolation does not cross them. The problem is one of how to define and find the important edges in the input image. Other edge-adaptive methods have been proposed by Jensen and Anastassiou [9], Li and Orchard

[12], and Muresan and Parks [15, 16, 17, 18]. The commercial software Genuine Fractals [2] also uses an edge-adaptive method to magnify images, but the details of the algorithm are not provided. Currently, the methods presented in [12, 18] are the most widely known edge-adaptive methods. They can well enough avoid jagged edges, but a limitation is that they sometimes introduce highly visible artifacts into the magnified images, especially in areas with small size repetitive patterns.

Yu *et al.* [25] presented a method that computes a triangulation of the input image where the pixels in the input image are the vertices of the triangles in the triangulation. The input image at any arbitrary scale is reconstructed by rendering its triangulation. However, since the edges in the input image are approximated using piecewise linear segments, curved edges cannot be properly reconstructed especially when the scaling factor is a large number.

Morse and Schwartzwald [14] presented a level-set reconstruction method to solve the problem of jagged edges. Their approach starts by magnifying the input image using the bicubic interpolation method, then iteratively smoothing the contours in the image. This approach, however, does not overcome the blurring problem found in the bicubic interpolation method.

Schultz and Stevenson [20] proposed a Bayesian approach to magnify images by hypothesizing the a priori probability. Atkins *et al.* [4], Hertzmann *et al.* [7], and Freeman *et al.* [6] proposed methods that learn the correspondences between low and high resolution images from a set of training data. The advantage of these approaches is that fine details can be added when producing high-resolution images when the input image is in the same class of image as the training data. The disadvantages of these approaches are that they will fail if the input image is not in the same class as the training data and that the computational cost is high.

Variational based approaches for image magnification have been presented by Malgouyres and Guichard [13] and Ballester *et al.* [5]. The magnified images obtained using these methods are better than those obtained using the bicubic interpolation method. However, since these methods solve optimization problems where all the pixels in the magnified image are unknowns, these methods, too, have high computational costs. As a result, they are not suitable for practical use.

We propose a progressive method for magnifying an image. In our method, we first create an initial magnified image, then we gradually refine the image until it becomes sharp. In the refinement process, the value of each pixel is refined by performing a local optimization. From experimental results, using the proposed method we are able to produce sharp magnified images without generating distinct artifacts.

3. Overview

Given an input image I of size $w \times h$ pixels and a scaling factor s (s is a real number), the problem is to create a magnified image I^s of size $\lfloor sw \rfloor \times \lfloor sh \rfloor$ pixels.

When s is a large number, generally it is better to magnify the input image gradually in order to produce a high quality magnified image. Therefore, our approach is as follows. Let $n = \lceil \log_2 s \rceil$. The magnified image I^s is created by doubling the size n times starting from I, producing an image I^{2^n} of size $2^n w \times 2^n h$ pixels, then scaling down I^{2^n} by the scaling factor $s/2^n$ using the bicubic interpolation method resulting in I^s.

In Section 4, we will describe our progressive approach for magnifying a gray scale image to twice its original size. Color images can be magnified by applying the proposed method to each color channel.

4. Progressive image magnification

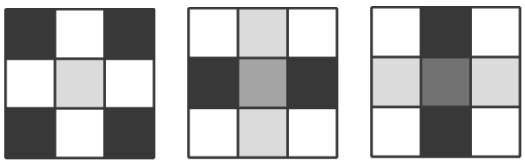
In this section, we describe a method to magnify a gray scale image to twice its original size. The gray level of a pixel is defined as a value between 0.0 and 1.0. Let the size of the input image I be $w \times h$ pixels. The size of the output image I^2 is $2w \times 2h$ pixels. We assume that the pixels at the locations $(2i, 2j)$, $(0 \le i \le w-1, 0 \le j \le h-1)$ of I^2 correspond to the pixels at the locations (i, j) of I. In the rest of this paper, we will use the term *original pixels* to refer to the pixels at $(2i, 2j)$ in the magnified image.

We magnify the input image in two steps. In the first step, an initial magnified image is created from the input image (Section 4.1) using a simple image magnification method that performs adaptive weighted average interpolation between the known pixel values. The result of this process is a magnified image that has sharper edges than the results of the other simple image magnification methods, for instance the bicubic interpolation method.

However, one can still distinguish that the initial magnified image is blurred compared to the input image. To create a high quality (sharp) magnified image, in the second step, the initial magnified image is progressively refined. For reasons that will become clear in the next section, the pixels at $(2i, 2j)$ in the initial magnified image usually do not have the same values as those of their corresponding pixels (i, j) in the input image. A sharp magnified image can be obtained by refining the initial magnified image until the values of these pixels match the values of their corresponding pixels in the input image (Section 4.2).

4.1. Initial magnified image

We have tested several simple image magnification methods, for instance, the bicubic interpolation and the cu-



(a) (b) (c)

Figure 1: (a) Computing the value of the pixel at $(2i+1, 2j+1)$ (the light gray pixel at the center) using the values at the four corner pixels (the black pixels). (b), (c) Computing the values of the pixels at $(2i+1, 2j)$ (the gray pixel at the center) and $(2i, 2j+1)$ (the dark gray pixel at the center) using the values at the four side pixels (the black and the light gray pixels).

bic convolution, for creating the initial magnified image. However, these methods tend to blur the edges in the image too much. In this section, we propose a simple image magnification method for doubling the size of the input image without blurring the edges too much. In our method, the initial magnified image is created in four steps.

In the first step, the values of the pixels at the locations $(2i, 2j)$, $(0 \leq i \leq w-1, 0 \leq j \leq h-1)$ are set by copying the values at pixels (i, j) in the input image.

In the second step, the values of pixels where the pixels at their four corners have values (pixels at the locations $(2i+1, 2j+1)$ as shown in Figure 1(a)) are computed by interpolating the values at the four corner pixels (the four black pixels). The values are computed as the weighted average of the values at the four corner pixels.

$$P(2i+1, 2j+1) = \frac{\sum_{(x,y)\in D} W(x,y)P(x,y)}{\sum_{(x,y)\in D} W(x,y)}, \quad (1)$$

where $P(x, y)$ and $W(x, y)$ are the value and the weight of the pixel at location (x, y), respectively, and $D = \{(2i, 2j), (2(i+1), 2j), (2i, 2(j+1)), (2(i+1), 2(j+1))\}$.

The weights of the corner pixels are computed as follows. $P(2i+1, 2j+1)$ is located between $P(2i, 2j)$ and $P(2(i+1), 2(j+1))$. If the difference between the values of $P(2i, 2j)$ and $P(2(i+1), 2(j+1))$ is small, then it is likely that there is no edge between these two pixels and their weights should be large. However, if the difference is large, then there is an edge between the two pixels. To preserve the sharpness of the edges, we do not want to interpolate across the edges. Therefore, their weights should be small. Let $d = P(2i, 2j) - P(2(i+1), 2(j+1))$. Considering the above, $W(2i, 2j)$ and $W(2(i+1), 2(j+1))$ are defined as $\exp(-d^2)$. The weights of the other two corner pixels are computed in the same manner.

In the third step, the values of pixels where the pixels at their four sides have values (pixels at the locations $(2i+1, 2j)$ as shown in Figure 1(b) and those at $(2i, 2j+1)$ as shown in Figure 1(c)) are determined by interpolating the values of the four side pixels (the black and the light gray pixels). The interpolation is performed using the same approach described in the second step.

At this point, all the pixels in the initial magnified image have values. However, there is a problem in that, in those regions where the colors should be uniform, the values of the pixels at $(2i, 2j)$ differ from the values of their surrounding pixels, resulting in color discontinuities. To solve this problem, in the fourth step, the values at pixels $(2i, 2j)$ are computed by interpolating the values of their eight neighbors. Again, the interpolation is performed using the same approach as the one in the second step.

4.2. Progressive refinement

Since the pixel values in the initial magnified image are computed by interpolating the values of several neighboring pixels, the initial magnified image is blurred compared to the input image, especially at edges.Our goal is to produce a magnified image that is as sharp as the input image.

As mentioned in Section 4.1, the final values of the pixels at $(2i, 2j)$ are computed by interpolating the values of their neighboring pixels. As a result, the values at such pixels usually do not match the values at their corresponding pixels (i, j) in the input image. Assume that the pixel value at (i, j) differs greatly from the pixel value at $(i+1, j)$ in the input image (that is, there is an edge between these two pixels). If we make the values of their corresponding pixels in the magnified image (the pixels at $(2i, 2j)$ and $(2(i+1), 2j)$) to match their values in the input image, then we can obtain a sharp edge between these pixels in the magnified image.

The problem is how to determine the values of the pixels other than the original pixels. To solve this problem, we use the original pixels to guide the transformation of the values of the rest pixels. The idea of our approach is to sharpen the initial magnified image by refining the image so that the values at the original pixels match their values in the input image. The refinement is performed progressively by transforming the value of each pixel in the magnified image, producing a new magnified image that has colors closer to the input image.

The value of each pixel is transformed as follows. Assume that we want to transform the value of pixel P at $(2i+1, 2j+1)$. Let p be the current value of this pixel and q be the new computed value of the pixel (q is still unknown). The four pixels at the corners of P (the four black pixels in Figure 1(a)), P_m $(m = 1, ..., 4)$ are the original pixels, that is the pixels which have the corresponding pixels in the input image. Let p_m and q_m be the value of P_m in the current magnified image and in the input image, respectively. Now, we have four pairs of values (p_m, q_m) $(m = 1, ..., 4)$ and a pair of values (p, q) where q is unknown.

As mentioned previously, one of the goal of the refinement process is to make the values of the original pixels to match the values of their corresponding pixels in the input image. When the refinement process is done we expect that the values of p_m are equal to q_m. This means that at the end of the refinement process, if we plot the pairs of values (p_m, q_m) in the 2D graph where the x and y axes representing the values of p_m and q_m, respectively, then the pairs (p_m, q_m) are located on a straight line $y = x$. This also means that during the refinement process, as the values of p_m change closer and closer to the values of q_m, then the pairs of values (p_m, q_m) $(m = 1, ..., 4)$ can be approximated using a straight line. Thus, we can write

$$q_m = \alpha p_m + \beta \ \ (m = 1, ..., 4). \tag{2}$$

Since P is located between P_m $(m = 1, ..., 4)$, it is reasonable to assume that point (p, q) is also located on the line expressed by Equation 2. As a result, we can compute q, which is the transformed value of p using Equation 2. The problem is how to compute the parameters α and β. It is obvious that in most cases, it is impossible to find a pair of values for α and β that satisfies Equation 2.

We solve this problem by using an optimization (least squares) approach. We define an error function E using the following equation.

$$E = \sum_m w_m (q_m - \alpha p_m - \beta)^2, \tag{3}$$

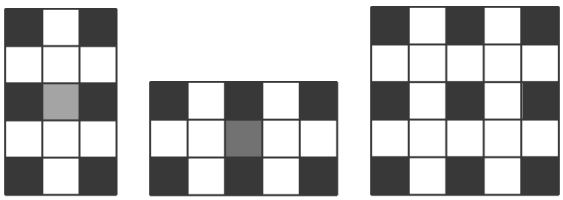
where $w_m = k(d_m)$ are the weights of the corner pixels ($k(u)$ is a non-negative B-spline kernel [23, 19] and d_m is the Euclidean distance between P and P_m). We compute the weight as a function of distance because the nearer a pixel is located to P the larger its weight should be. Since we do not want negative weights, we choose the non-negative B-spline kernel from among the existing interpolation kernels (most of the interpolation kernels, for instance the cubic convolution kernel [23], have negative lobes). The parameters α and β are computed such that the value of E is minimized. This is achieved by solving the following system of linear equations containing the equations $\partial E/\partial \alpha = 0$ and $\partial E/\partial \beta = 0$.

$$\sum_m w_m {p_m}^2 \alpha + \sum_m w_m p_m \beta = \sum_m w_m q_m p_m. \tag{4}$$

$$\sum_m w_m p_m \alpha + \sum_m w_m \beta = \sum_m w_m q_m. \tag{5}$$

The approach mentioned above will fail if all the values of p_m are close to each other since the determinant of the system of linear equations will be near to zero, resulting in a considerable possibility of numerical error. In this kind of case, we set α to one and use the following equation to define the approximation.

$$q_m = p_m + \beta \ \ (m = 1, ..., 4). \tag{6}$$



(a) (b) (c)

Figure 2: The original pixels (the black pixels) used for refining the values of the pixels at (a) $(2i + 1, 2j)$ (the gray pixel), (b) $(2i, 2j + 1)$ (the dark gray pixel), and (c) $(2i, 2j)$ (the black pixel at the center).

Then, the error function E becomes

$$E = \sum_m w_m (q_m - p_m - \beta)^2. \tag{7}$$

The optimal value of β is

$$\beta = \frac{\sum_m w_m (q_m - p_m)}{\sum_m w_m}. \tag{8}$$

Using the computed parameters, the value p at pixel P is transformed to a new value q. Let $q_{max} = \max\{q_m\}$ and $q_{min} = \min\{q_m\}$. Since P is located between P_m $(m = 1, ..., 4)$, we make sure that its new value q is between the values of q_{max} and q_{min}. If q is larger than q_{max}, then q is set to q_{max} whereas if q is smaller than q_{min}, then q is set to q_{min}.

The values of the rest of the pixels are transformed in the same manner. The values of the pixels at the locations $(2i+1, 2j)$ and $(2i, 2j+1)$ are transformed using the nearest six original pixels, that is the six black pixels in Figures 2(a) and (b), respectively. The original pixels $(2i, 2j)$ are transformed using the nearest eight original pixels and itself, that is the nine black pixels in Figure 2(c). We have tested transformation of the values of the original pixels directly to their values in the input image. In this case, the refining process is performed only once. However, from experimental results, this approach produces magnified images that are not very sharp compared to the input images. When refining the value of an original pixel using itself and its neighbors, its own value is assigned the largest weight compared to its neighboring pixels, and as a result, its refined value will get closer and closer to the value of its corresponding pixel in the input image.

The refining process is performed until the differences between the values of the original pixels, the pixels at $(2i, 2j)$ in the magnified image, and the values of their corresponding pixels, the pixels at (i, j) in the input image, are below a certain threshold. From experimental results, in most cases the refining process is performed for less than ten iterations.

(a) the input image

(b) the initial magnified image

(c) the result after one iteration

(d) the result after four iterations

(e) the final magnified image

Figure 3: Magnifying a butterfly image to twice its original size. (a) The input image, (b) the initial magnified image, (c) the refined image after one iteration, (d) the refined image after four iterations, and (e) the final magnified image after eight iterations.

5. Results and discussions

Figure 3 shows the result of doubling the size of a butterfly image (Figure 3(a)). The size of the input image is 216×216 pixels. Figure 3(b) shows the initial magnified image. Figures 3(c) and (d) show two intermediate images during the refinement process. Figure 3(e) shows the final magnified image obtained after eight refinement iterations. It is clear that the edges, for instance the edges near the antenna of the butterfly, in the magnified image after the refinement process are sharper compared to the edges in the initial magnified image.

Next, we applied our method to three input images in order to examine the convergence behavior of the proposed refinement algorithm, We doubled the size of three input images, the butterfly image (Figure 3(a)), the Barbara image (Figure 6(a)), and the mandrill image (Figure 6(b)). The sizes of the Barbara and the mandrill images are 512×512 pixels.

Since we check convergence by comparing the values of the original pixels and their corresponding pixels in the input image, to examine the convergence behavior, we plot a graph that shows the changes of the maximum differences, during the refinement process, between the values of the original pixels in the magnified images and their corresponding pixels in the input images. The graph in Figure 4 is the plotting result when we applied our method to the three test images. Please note that we define the gray level

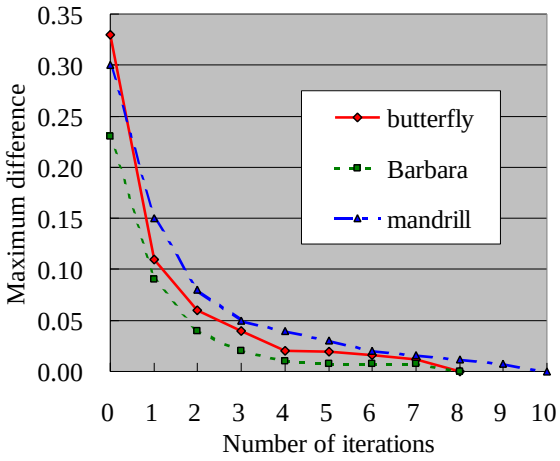


Figure 4: The graph showing the changes of the maximum differences, during the refinement process, between the values of the original pixels in the magnified images and their corresponding pixels in the input images.

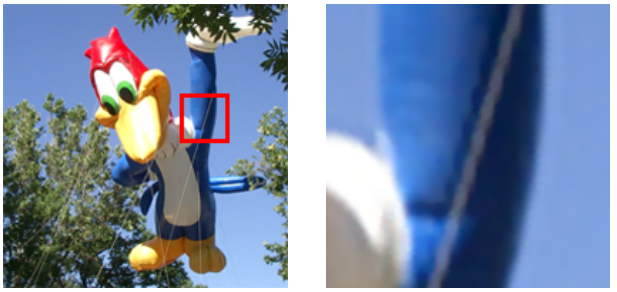
of a pixel as a value between 0.0 and 1.0, and thus the values of differences between pixels are in the range of 0.0 to 1.0.

We observed that our refinement algorithm behaved similarly for all the three test images. The initial magnified images (when the iteration is zero) have the largest difference values. Generally, the pixels near the edges in the image have large difference values. After each refinement step, the maximum differences decreased to almost half the values at the previous step. After several iterations, the differences are so small, as a result, we can get magnified images that have almost the same visual quality as the final magnified images. For instance, in the case of the butterfly image, the result after four iterations (Figure 3(d)) have almost the same visual quality as the final result after eight iterations (Figure 3(e)).

As for the computational times, the two times magnifications took about six seconds for the butterfly image and about thirty seconds for the Barbara and the mandrill images. The computations were performed on a machine with 2.0 GHz Xeon.

5.1. Comparison to other methods

Figure 5 compares the results of the proposed method and the bicubic interpolation method when magnifying the butterfly image by a factor of four. We used the bicubic interpolation function in Adobe Photoshop [1] in the comparison. For the comparison, we only show some portions of the magnified image. One can observe that using the proposed method, we can produce magnified images with sharper edges than the bicubic interpolation method.



(a) mascot (b) four times magnification

Figure 9: Magnifying (a) a mascot image four times its original size. (b) Jagged straight line can be seen in the region of the magnified image that corresponds to the region inside the red rectangle shown in (a).

We also performed some comparisons to a more elaborate method. There are many methods proposed on image magnification as mentioned in Section 2. From a practical view, the edge directed methods are the state-of-the-art technologies of the image magnification methods. They are able to produce smoother and sharper edges in the magnified images compared to other approaches and they also do not depend on learning sets. Therefore, we chose the edge directed method for performing the comparison. We compared our results and the results of the latest edge directed method proposed by Muresan and Parks [18]. They have proposed several image magnification methods and the method in [18] produces the best results. We also chose this method because it generally produces better results than the other edge directed interpolation methods.

Figure 6 shows the input images used for the comparison. The input images were magnified by a factor of four. As shown in Figure 7(a), highly visible artifacts were produced in the result using the method in [18]. On the other hand, using our method, we produced a magnified image (Figure 7(b)) with sharper and better quality edges. Figures 7(c) and (d) show the results when both methods were applied to magnify a mandrill image containing fine textures. In the areas of fine textures (areas under the eye), it is easier to distinguish the different regions in the image created using our method (Figure 7(d)) than the one created using the method in [18] (Figure 7(c)).

5.2. Magnification of color images

The magnification of a color image is performed by independently applying the proposed method to each color channel. Figure 8 shows the result of magnifying a color image which is an RGB image. The size of the input flower image (Figure 8(a)) is 200×200 pixels. Figure 8(b) shows the result of magnifying this flower image by a scaling factor of eight.

5.3. Limitations

The method used to compute the initial magnified image does not guarantee the smoothness of the edges in the resulting image. As a result, jagged edges can appear in the final magnified image (Figure 9). In order to further increase the quality of the magnified image, we would like to incorporate the level-set reconstruction method [14] to smooth the edges before performing the refinement process.

6. Conclusions and future work

In this paper, we have presented a novel approach for doubling the size of an input image. The important contribution of this paper is a method to sharpen a magnified image guided by its corresponding low-resolution image using the optimization approach.

To magnify an input image, an initial magnified image is created by using a simple magnification method that considers the derivatives between the pixel values when computing the weights of the pixels during interpolation. The initial magnified image is then progressively refined until the colors match those of the input image, producing a high quality magnified image. Employing the proposed method, we are able to produce magnified images that have sharp edges without introducing distinct artifacts.

Interesting future work is to extend the proposed method to deal with the problem of increasing the resolution of other types of data such as 3D volume data and 2D/3D vector fields.

References

[1] *Adobe Photoshop*. Adobe Systems.

[2] *Genuine Fractals*. LizardTech.

[3] J. Allebach and P. W. Wong. Edge-directed interpolation. In *Proceedings of IEEE International Conference on Image Processing Vol. 3*, pages 707–710, 1996.

[4] C. B. Atkins, C. A. Bouman, and J. P. Allebach. Optimal image scaling using pixel classification. In *Proceedings of IEEE International Conference on Image Processing Vol. 3*, pages 864–867, 2001.

[5] C. Ballester, V. Caselles, and J. Verdera. Disocclusion by joint interpolation of vector fields and gray levels. *Multiscale Modelling and Simulation*, 2(1):80–123, 2003.

[6] W. T. Freeman, T. R. Jones, and E. C. Pasztor. Example-based super-resolution. *IEEE Computer Graphics and Applications*, 22(2):56–65, 2002.

[7] A. Hertzmann, C. E. Jacobs, N. Oliver, B. Curless, and D. H. Salesin. Image analogies. In *Proceedings of SIGGRAPH 2001*, pages 327–340, 2001.

[8] H. S. Hou and H. C. Andrews. Cubic splines for image interpolation and digital filtering. *IEEE Transactions on Acoustics, Speech, Signal Processing*, 26(6):508–517, 1978.

[9] K. Jensen and D. Anastassiou. Subpixel edge localization and the interpolation of still images. *IEEE Transactions on Image Processing*, 4(3):285–295, 1995.

[10] R. Keys. Cubic convolution interpolation for digital image processing. *IEEE Transactions on Acoustics, Speech, Signal Processing*, 29(6):1153–1160, 1981.

[11] S. W. Lee and J. K. Paik. Image interpolation using adaptive fast B-spline filtering. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing Vol. 5*, pages 177–179, 1993.

[12] X. Li and M. T. Orchard. New edge-directed interpolation. *IEEE Transactions on Image Processing*, 10(10):1521–1527, 2001.

[13] F. Malgouyres and F. Guichard. Edge direction preserving image zooming: a mathematical and numerical analysis. *SIAM Journal on Numerical Analysis*, 39(1):1–37, 2001.

[14] B. S. Morse and D. Schwartzwald. Image magnification using level-set reconstruction. In *Proceedings of IEEE International Conference on Computer Vision*, pages 333–341, 2001.

[15] D. D. Muresan and T. W. Parks. New image interpolation techniques. In *Proceedings of IEEE 2000 Western New York Image Processing Workshop*, 2000.

[16] D. D. Muresan and T. W. Parks. Adaptive, optimal-recovery image interpolation. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing Vol. 3*, pages 1949–1952, 2001.

[17] D. D. Muresan and T. W. Parks. Optimal recovery approach to image interpolation. In *Proceedings of IEEE International Conference on Image Processing Vol. 3*, pages 848–851, 2001.

[18] D. D. Muresan and T. W. Parks. Adaptively quadratic (AQua) image interpolation. *IEEE Transactions on Image Processing*, 13(5):690–698, 2004.

[19] M. Salisbury, C. Anderson, D. Lischinski, and D. H. Salesin. Scale-dependent reproduction of pen-and-ink illustration. In *Proceedings of SIGGRAPH 96*, pages 461–468, 1996.

[20] R. R. Schultz and R. L. Stevenson. A Bayesian approach to image expansion for improved definition. *IEEE Transactions on Image Processing*, 3(3):233–242, 1994.

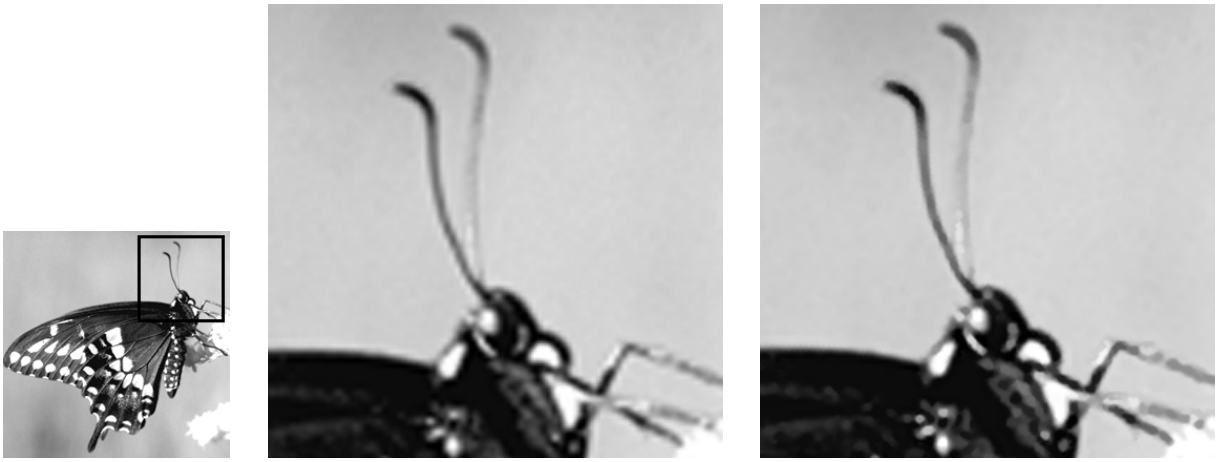
[21] S. Thurnhofer and S. Mitra. Edge-enhanced image zooming. *Optical Engineering*, 35(7):1862–1870, 1996.

[22] M. Unser, A. Aldroubi, and M. Eden. Fast B-spline transforms for continuous image representation and interpolation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(3):277–285, 1991.

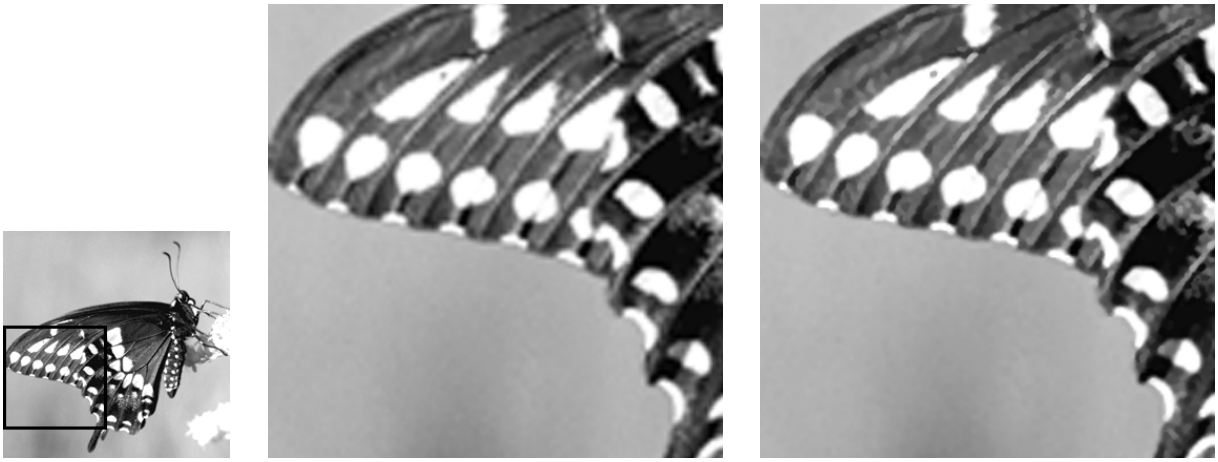
[23] G. Wolberg. *Digital image warping*. IEEE Computer Society Press, 1990.

[24] G. Wolberg and H. Massalin. A fast algorithm for digital image scaling. In *Proceedings of Computer Graphics International '93*, 1993.

[25] X. Yu, B. S. Morse, and T. W. Sederberg. Image reconstruction using data-dependent triangulation. *IEEE Computer Graphics and Applications*, 21(3):62–68, 2001.



(a) the input image (b) the result using the bicubic interpolation (c) the result using the proposed method

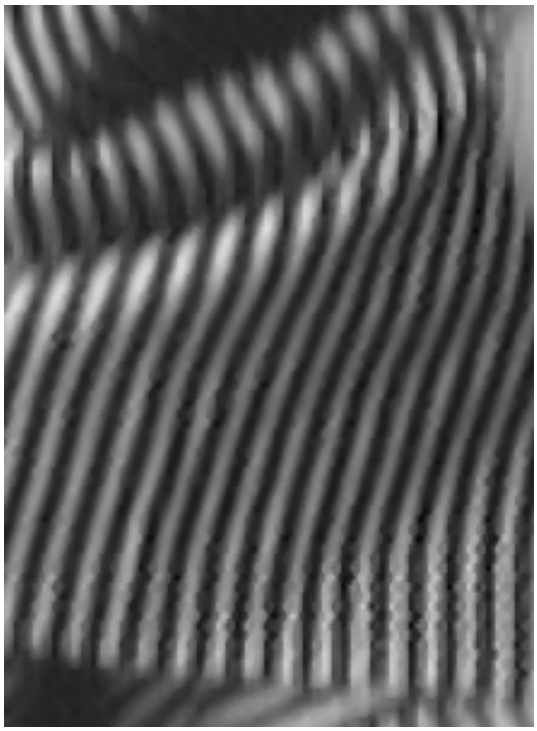


(d) the input image (e) the result using the bicubic interpolation (f) the result using the proposed method

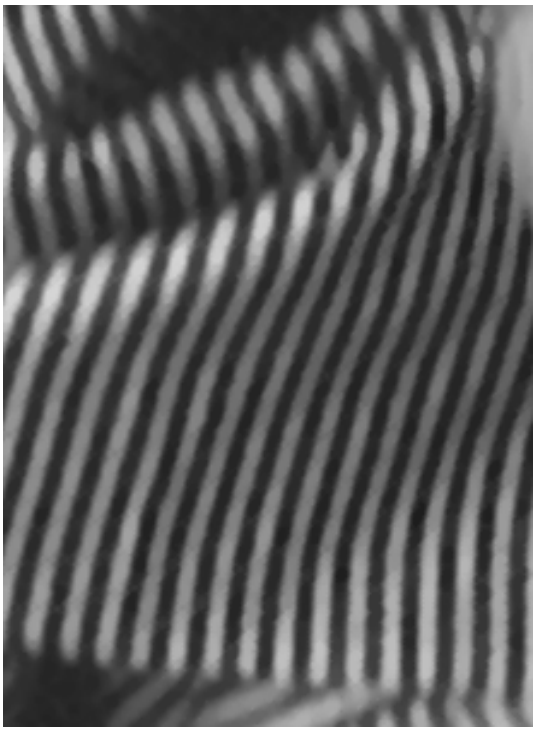
Figure 5: Magnifying the butterfly image four times. (a) and (d) are the input images, (b) and (e) are the results obtained using the bicubic interpolation function in Adobe Photoshop [1], (c) and (f) are the results obtained using our method.

(a) Barbara (b) mandrill

Figure 6: The images used for comparing the proposed method and the method of Muresan and Parks [18]. These images are magnified four times and Figure 7 shows the portions of the magnified images that correspond to the regions inside the black rectangles shown here.



(a) the magnified Barbara image using the method in Muresan and Parks [18]



(b) the magnified Barbara image using the proposed method

(c) the magnified mandrill image using the method in Muresan and Parks [18]

(d) the magnified mandrill image using the proposed method

Figure 7: Magnifying the Barbara and the mandrill images in Figure 6 by a factor of four. (a) and (c) are the results obtained using the method proposed by Muresan and Parks [18], (b) and (d) are the results obtained using our method.

(a) the input image

(b) the magnified image

Figure 8: Magnifying a flower image eight times.