Fast Artifacts-Free Image Interpolation

Andrea Giachetti[1] and Nicola Asuni[2]
[1] Dipartimento di Informatica, Università degli Studi di of Verona,
Strada Le Grazie 15, 37134 Verona, Italy
andrea.giachetti@univr.it
[2]Tecnick.com, Via Della Pace, 11, 09044 Quartucciu (CA), Italy

Abstract

In this paper we describe a novel general purpose image interpolation method based on the combination of two different procedures. First, an adaptive algorithm is applied interpolating locally pixel values along the direction where second order image derivative is lower. Then interpolated values are modified using an iterative refinement minimizing differences in second order image derivatives, maximizing second order derivative values and smoothing isolevel curves. The first algorithm itself provides edge preserving images that are measurably better than those obtained with similarly fast methods presented in the literature. The full method provides interpolated images with a "natural" appearance that do not present the artifacts affecting linear and nonlinear methods. Objective and subjective tests on a wide series of natural images clearly show the advantages of the proposed technique over existing approaches.

1 Introduction

Image upscaling is often necessary for a variety of applications, such as printing, video streaming, image processing, texture mapping for model rendering, etc. Simple techniques like pixel replication or linear interpolation are not satisfactory due to the creation of visual artifacts like pixelization, jagged contours, oversmoothing. To overcome these problems, new algorithms creating upscaled images with sharper edges and reduced artifacts have been introduced, but an optimal method providing reasonably natural and artifact free upscaled images with a low computational complexity did not emerge clearly.

In this paper we propose a new general purpose image upscaling method which uses local second order information to adapt the interpolation and an iterative refinement able to remove artifacts while preserving relevant image features and natural texture.

Objective and subjective tests against other well known methods show that the method is clearly an improvement on previous ones. It provides a reasonable reconstruction of the missing information and requires considerably less computational power than other method achieving good scores, a feature that is also extremely important for interpolation algorithms, especially in the case of real time applications. The paper is organized as follows: Section 2 presents an overview of different image upscaling methods, Section 3 describes the new methods proposed, Section 4 discusses experimental results.

2 Interpolation techniques: a review

Several approaches have been proposed to guess reasonable high resolution patterns from low resolution original images. The problem is quite complex as, in general, no hints on the real high resolution signal are available. The most sophisticated interpolation methods try to extract statistical information on the relationship between high and low resolution images from a training set of natural images (or images of interest). A similar approach is used in [8], where texton substitution is used to increase the resolution of image sequences. In [7] a Markov Model is used to optimize patch replacement preserving continuity. A similar approach is also used in [15], where "primal sketch priors" are constructed and used to enhance the quality of the high resolution images. In [6] the parameters of an edge model are related to the low resolution patches using machine learning. In [2] low resolution pixels are first classified in the context of a window of neighboring points, then different filters are used depending on the classification results. The problem with this approach is that the huge variety of natural textures and scales makes a general purpose use of the techniques quite difficult, though they can be efficiently applied to particular tasks (i.e. search of patterns such as faces, trees, etc.).

The interpolation methods usually applied in practical applications do not use statistical information and make only smoothness and edge continuity assumptions to guess reasonable high resolution patterns. Despite the simplicity of the task and the intuitive assumptions, however, a variety of methods have been presented to achieve this goal and a dominant method unanimously adopted for the task has not clearly emerged. The simplest algorithms compute the interpolated values as a linear combination of the original ones closest to the new position. Using different neighborhoods and kernels, the classic bilinear, bicubic, Lanczos and other methods widely applied in image viewers and image processing tools have been obtained. These methods are computationally efficient, even if the images obtained do not appear "natural" due to excessive blurring and jagged artifacts.

Nonlinear methods have been proposed to overcome these limits, usually exploiting local features like edges (they are often called edge-adaptive). In [10] the interpolation is guided by the output of directional filter banks. In [13] the high resolution image is modeled as a Gibbs-Markov Field and the zooming procedure is obtained optimizing convex functionals. In [16] a Laplacian Pyramid decomposition is performed and used for the prediction of local high frequency components. The approach developed in [12] consists of first determining the local quadratic signal from local patches, then estimating missing samples applying optimal recovery. In [11] an iterative method based on level set theory and isophotes (i.e. curves of constant intensity) smoothing is applied, along with some ad hoc rules to prevent change in topology and other side effects. In [17] a kernel that adapts to the local orientation of isophotes is used to reduce artifacts in bilinear interpolation. Also in [4], the authors use bilinear interpolation and then try to correct the error by utilizing the interpolation error theorem in an edge-adaptive way.

A particular class of efficient edge-based algorithms is represented by methods using simple heuristics to determine the edge direction from 4 neighboring points in the low resolution image and then add interpolated pixels obtained as weighted averages of the neighbors, with weights depending on the edge direction. Algorithms of this kind are the Data Dependent Triangulation [14] and the methods proposed in [3] and [5]. A similar approach is also the basis of the NEDI technique [9] that uses a large patch around the 4 points to estimate the weights, assuming local constant covariance at different scales. We have shown in a recent paper [1] that, using adaptive windows and other simple heuristics, it is possible to improve the performances of the original NEDI implementation obtaining very good results, even in the case of large scale factors. Unfortunately we found two major drawbacks in the method: high computational complexity (several minutes to enlarge common images) and creation of evident artifacts near high frequency patterns, due to the large support required for the least squares covariance estimation. We were therefore motivated to find a method that could combine the efficiency of the fast edge directed interpolators and the image quality of the improved NEDI.

3 Interpolation based on second order image derivatives

Methods described at the end of the previous section, enlarge images approximately by a factor of 2 by copying original pixels (indexed by i, j) into an enlarged grid (indexed by 2i, 2j) and then filling gaps with ad hoc rules based on local edge analysis (larger magnification factors can be obtained by recursively applying the algorithm on the output).

The procedure is usually performed in two steps: in the first one, pixels indexed by two odd values (darker pixel in Fig. 1 A) are computed as a weighted average of the four diagonal neighbors (pixels of the original image); in the second the remaining holes (e.g. black pixel in Fig. 1 B) are filled with the same rule, as a weighted average of the 4 nearest neighbors (in horizontal and vertical directions).

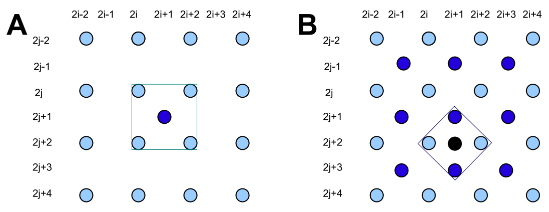

Figure 1: Two step interpolation based on the configuration of four neighbors

The NEDI method [9] computes the weights by assuming the local image covariance constant in a large window and at different scales. With this constraint, an overconstrained equation system can be obtained and solved. Using some tricks to adapt window size and to handle matrix conditioning , we obtained a modified (iNEDI) method see [1], providing sharp images although with a high computational cost and a rather unnatural texture in high frequency regions.

Fast methods using similar pixel replication and hole filling approaches are extremely efficient, even if the quality of the results is not comparably good. In [14], the central point is obtained interpolating the two opposite ones in the direction where the pixel difference is lower. In [3, 5] more complex heuristics are used, still based on pixel differences and thresholding. These methods are interesting due to the edge adaptivity and the low complexity, but they fail to capture the correct edge behavior in several cases. Even if they provide a visual appearance with less jagged artifacts they usually perform worse than bicubic interpolation in objective comparisons. We believe that a big limitation of these methods is that they just try to minimize image variation (first order derivatives) along selected directions, while NEDI interpolation is better adapted to the most probable edge shape because it use second order information to obtain the coefficients. In fact, the constant covariance condition, if the local image variation is not too big, is equivalent to the assumption of local constancy of second order derivatives, roughly approximating curvatures of the profile describing image brightness along the selected direction.

Focusing on this idea, we designed a new interpolation method composed of two different procedures. The first is a simple rule-based hole-filling method which computes new samples by interpolating along the direction where the second order image derivative is lower (FCBI, Fast Curvature Based Interpolation). The second refines the values of the interpolated pixels through an iterative algorithm trying to force second order derivative continuity with some additional constraints for edge preservation. The result is a new interpolation algorithm that we call ICBI (Iterative Curvature Based Interpolation).

3.1 Fast curvature based interpolation

The method is similar to the Data Dependent Triangulation, but instead of obtaining the new pixel values by averaging the two opposite neighbors with lower difference, we compute second order derivatives in the two diagonal directions and interpolate the two opposite neighbors in the direction where the derivative is lower. In detail, if we consider the first interpolation step shown in Fig. 1 A, we compute the local approximation of the second order derivatives $\tilde{I}_{11}(2i+1,2j+1)$ and $\tilde{I}_{22}(2i+1,2j+1)$ along the two diagonal directions using a 12 pixel neighborhood (2) as:

$$\begin{aligned}\tilde{I}_{11}(2i+1,2j+1) &= I(2i-2,2j+2)+I(2i,2j)+I(2i+2,2j-2) \qquad (1)\\ &\quad -3I(2i,2j+2)-3I(2i+2,2j)+I(2i,2j+4)+I(2i+2,2j+2)+I(2i+4,2j)\\ \tilde{I}_{22}(2i+1,2j+1) &= I(2i,2j-2)+I(2i+2,2j)+I(2i+4,2j+2)+ \qquad (2)\\ &\quad -3I(2i,2j)-3I(2i+2,2j+2)+I(2i-2,2j)+I(2i,2j+2)+I(2i+2,2j+4)\end{aligned}$$

The interpolated value at location (2i+1,2j+1) is then:

$$\begin{array}{ll} (I(2i,2j)+I(2i+2,2j+2))/2 & \text{if } \tilde{I}_{11}(2i+1,2j+1) > \tilde{I}_{22}(2i+1,2j+1), \\ (I(2i+2,2j)+I(2i,2j+2))/2 & \text{otherwise.} \end{array}$$

The second step is performed in the same way, computing the approximations of the second order derivatives in horizontal and vertical directions. We only avoid the use of this simple rule when first order derivatives of the intensity are larger than a fixed, large

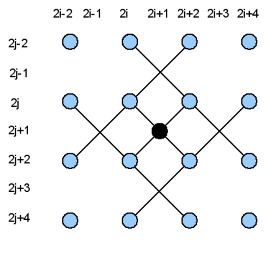

Figure 2: Edge-based interpolation based on a 12-pixels neighborhood: the largest second order diagonal derivative determines the interpolation direction.

threshold (in this case the derivative is not a good approximation of directional curvature). In this case we interpolate in the direction with lower gray level difference.

Images obtained with this fast method are better than those obtained with similar fast edge directed methods (see Section 4), but not comparable, for example, with that obtained with improved NEDI. We therefore improved the algorithm by adding an iterative refinement able to improve edge quality at each interpolation step by smoothing second order directional derivatives keeping original pixel values fixed and adding constraints to preserve sharp discontinuities. The procedure is motivated by the fact that it is easy to derive, from NEDI equations, that one of the effects of the constant covariance constraint is, in case of limited intensity differences, to impose local continuity in second order derivatives of the intensity itself.

3.2 Iterative curvature based interpolation

In the following, we describe the procedure and the energy terms used for the first interpolation step (filling gaps in the enlarged grid at locations $(2i+1,2j+1)$). Modified formulas for the second step filling the remaining gaps are straightforward. First we introduced the term:

$$\begin{aligned} U_c(2i+1,2j+1) = \\ & w_1(|(I_{11}(2i,2j)-I_{11}(2i+1,2j+1))|+|(I_{22}(2i,2j)-I_{22}(2i+1,2j+1))|)+ \\ & w_2(|(I_{11}(2i,2j)-I_{11}(2i+1,2j-1))|+|(I_{22}(2i,2j)-I_{22}(2i+1,2j-1))|)+ \\ & w_3(|(I_{11}(2i,2j)-I_{11}(2i-1,2j+1))|+|(I_{22}(2i,2j)-I_{22}(2i-1,2j+1))|)+ \\ & w_4(|(I_{11}(2i,2j)-I_{11}(2i-1,2j-1))|+|(I_{22}(2i,2j)-I_{22}(2i-1,2j-1))|) \end{aligned} \quad (3)$$

where I_{11}, I_{22} are local approximations of second order directional derivatives, computed as:

$$I_{11}(2i+1,2j+1) = I(2i-2,2j-2)+I(2i+2,2j+2)-2I(2i+1,2j+1) \quad (4)$$

$$I_{22}(2i+1,2j+1) = I(2i-2,2j+2)+I(2i+2,2j-2)-2I(2i+1,2j+1) \quad (5)$$

This energy term sums local directional changes of second order derivatives. Weights w_i are set to 1 when the first order derivative in the corresponding direction is not larger than a fixed threshold and to 0 otherwise. In this way smoothing is avoided when there is a strong discontinuity. The minimization of this term roughly corresponds to force continuity in the local curvatures of the continuous surface ideally describing the image. It tends to keep the local curvatures at different scales constant, as the constant covariance constraint does in the NEDI method. The only problem of the method can be an excessive smoothing, which can be reduced by adding other energy terms. We therefore added an energy term enhancing the absolute value of the second order derivative:

$$U_e(2i+1,2j+1) = -|I_{11}(2i+1,2j+1)|+I_{22}(2i+1,2j+1)| \quad (6)$$

and a term for isophotes (i.e. isolevel curves) smoothing. This is derived from [11] who used for an iterative isophote smoothing a local force defined as

$$f(I) = -\frac{I_1(i,j)^2 I_{22}(i,j) - 2I_1(i,j)I_2(i,j)I_{12}(i,j) + I_{22}(i,j)^2 I_1(i,j)}{I_1(i,j)^2 + I_2(i,j)^2}) \quad (7)$$

with $I_{11}, I_{22}, I_{12}, I_1, I_2$ being local approximations of first and second order directional derivatives.

This force alone was not able to produce very good and natural interpolated images, as also stated in the original paper, while the addition of a derived potential to the curvature continuity and curvature enhancement terms is able to improve the quality of our interpolation results in edge regions. The potential is defined as:

$$U_i(2i+1,2j+1) = f(I)|_{2i+1,2j+1} I(2i+1,2j+1) \tag{8}$$

I_{11}, I_{22} are computed as before, the other derivatives estimates are obtained as:

$$I_{12}(2i+1,2j+1) = \tag{9}$$
$$0.5(I(2i+1,2j-1) + I(2i+1,2j+3) - I(2i+1,2j+1) - I(2i+3,2j+1))$$
$$I_1(2i+1,2j+1) = 0.5(I(2i,2j) - I(2i+2,2j+2)) \tag{10}$$
$$I_2(2i+1,2j+1) = 0.5(I(2i,2j+2) - I(2i+2,2j)) \tag{11}$$

The complete energy function for each pixel location $(2i+1,2j+1)$, sum of the "curvature continuity", "curvature enhancement" and "isophote smoothing" terms becomes therefore:

$$U(2i+1,2j+1) = \alpha U_c(2i+1,2j+1) + \beta U_e(2i+1,2j+1) + \gamma U_i(2i+1,2j+1) \tag{12}$$

The first step of the iterative interpolation correction (adjusting pixel values with two odd indexes) is finally implemented as a simple greedy minimization of the energy: after the placement of the original pixels at locations $(2i,2j)$ and the insertion of rough interpolated ones at locations $(2i+1,2j+1)$, we compute, for each new pixel, the energy function $U(2i+1,2j+1)$ and the two modified energies $U^+(2i+1,2j+1)$ and $U^-(2i+1,2j+1)$, i.e. the energy values obtained by adding or subtracting a fixed value δ to the the local pixel value $I(2i+1,2j+1)$. The intensity value corresponding to the lower energy is then assigned to the pixel. This procedure is iteratively repeated until the sum of the modified pixels at the current iteration is lower than a fixed threshold, or the maximum number of iterations has been reached. The number of iterations can be also fixed in order to adapt the computational complexity to timing constraints. α, β and γ were chosen by trial and error to obtain a tradeoff between edge sharpness and artifacts removal.

After the second hole-filling step (assigning values to all the remaining empty pixels), the iterative procedure is repeated in a similar way, just replacing the diagonal derivatives in the energy terms with horizontal and vertical ones and iteratively modifying only the values of the newly added pixels. The complete ICBI algorithm is summarized in Fig. 3.

4 Experimental results

We tested the algorithms proposed on a database of 25 natural images selected from the morgueFile online archive (http://morguefile.com). All images are subject to the license agreement available at the web page http://morguefile.com/archive/terms.php. Selected images (RGB, 8 bit) provide a wide range of colors and natural textures. In the algorithm description the equation are written for grayscale images; of course, color images can be enlarged in the same way by repeating the procedures independently for each color component or computing interpolation coefficients on the image brightness and using them for all the channels, reducing in this way the computational cost and avoiding color artifacts.

```
• Put original pixels in the enlarged grid at locations 2i,2j
• Insert pixels at locations 2i+1 2j+1 with the FCBI method
• Apply iterative correction until the image variation is above a given
  threshold or the maximum number of iterations is reached
• Insert pixels in the remaining locations with the FCBI method
• Apply iterative correction to the added pixels
• Repeat the whole procedure on the new image for further enlargements
```

Figure 3: Pseudocode for the ICBI algorithm

Figure 4: Selected images from the test database.

4.1 Objective test

The objective test compares images obtained downsampling the original images and then enlarging them with different methods with reference images obtained just downsampling the original ones to the corresponding size. For simplicity we converted images to 8 bit grayscale, the use of all three color channel being not relevant to this test. We created 128x128 and 256x256 subsampled version of the original images and the different downsampled reference images correctly shifted in order to compensate the slightly different zoom factors and translation created by the algorithms. In fact, methods like ours do not provide exact 2X or 4X scale factors, being the exact enlargement at each step $(2width-1)/width$ horizontally and $(2height-1)/height$ vertically. Finally, we enlarged the 256x256 images by a 2X factor and the 128x128 versions by a 4X factor, and measured the differences between the upscaled images and the reference ones by evaluating the Peak Signal to Noise Ratio, defined as:

$$PSNR = 20\log_{10} \frac{MAXPIX}{\frac{\sum_{i=1}^{W}\sum_{j=1}^{H}(I_{up}(i,j)-I_{orig}(i,j))^2}{(W*H)}} \tag{13}$$

where $I_{up}(i,j)$ is the upscaled subsampled image, I_{orig} the original one, W and H the image dimensions and $MAXPIX$ the end scale value of the pixel intensity.

The results for a 2X enlargement of 256x256 images and 4X enlargement of 128x128 images are summarized in Table 1. The results for FCBI and ICBI have been obtained with a Matlab implementation of the proposed algorithms, Nearest Neighbor and Bicubic interpolated images have been obtained with Matlab builtin functions, original NEDI and iNEDI interpolated images were obtained with Matlab implementations and Chen's method was implemented by us according to [5].

The iterative method produced clearly the best results, but the most interesting fact is that the iterative method is relevantly faster than iNEDI. Table 2 shows the average time necessary to enlarge images with different methods. Computation times reported in tables

	ICBI	FCBI	iNEDI	NEDI	Chen	Iso.	Bic.	NN
2X dB	**31.07**	29.82	30.64	29.71	29.50	29.47	30.36	28.13
4X dB	**25.33**	24.46	25.18	24.30	24.19	24.11	24.91	23.43

Table 1: Peak signal to noise ratios (dB) obtained by comparing images upscaled by approximately 2X and 4X factors with reference images. The new iterative method provided the best results, while the FCBI method produced images closer to the reference ones than those obtained with similar methods based on first order derivatives (i.e. Chen's).

	ICBI	FCBI	iNEDI	NEDI	Chen
2X time(s)	12.91	0.17	312.44	221.64	0.11
4X time(s)	13.30	0.18	372.83	293.36	0.12

Table 2: Average computation times obtained with non-optimized Matlab implementations of the algorithms. A C++ implementation of the ICBI methods requires about 1.2 sec. for 4x upscaling.

are obtained with non optimized Matlab implementations on a Dell XPS M1210 laptop with an Intel Core2 Duo T7200 2.0 GHz CPU. The iterative nature of the algorithm can be also used to adapt image quality to the available hardware performances. By limiting the number of iterations we can obtain good quality and artifacts-free images with a reduced computational cost: with a maximum of 5 iterations, for example, the PSNR differ on average less than 0.1 dB from the best value and the computation is three times faster. In order to test the possibility of reaching real time performances, we ported ICBI in C++, using the OpenCV libraries for image handling and display (Gentoo Linux, same hardware as before). The time required for a 4X upscaling of a 128x128 images has been reduced to an average of 1.2 sec. (same parameters of Matlab tests in Table 2).

4.2 Subjective test

In order to compare perceived image quality, we have taken a subset of 10 of the previously described RGB images and enlarged them by a 4x factor with six different algorithms (NEDI, iNEDI, bicubic, Chen's, FCBI, ICBI). We then asked a group of 12 people to compare them in order to select the method providing the best average "perceived quality". All the different possible couples of corresponding images were presented (in random order) to the subjects involved in the test, who were asked to choose the preferred image for each of them. The sum of the successful comparisons for each interpolation method was then taken as the quality score of the method itself. The average scores (total number of preferences divided by the number of images multiplied by the number of subjects) are reported in Fig. 5. We can observe some differences between the results obtained and the results of the objective test. FCBI is now preferred to bicubic interpolation being able to remove jagged artifacts that strongly affect bicubic interpolation. Images enlarged with ICBI were usually rated as the best. It must be considered that the only method with scores not too far from ICBI is iNEDI a method presenting a prohibitive computational complexity for most practical applications. Fig 6 shows the evident artifact removal obtained with the ICBI method.

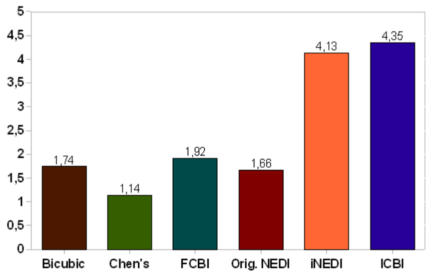

Figure 5: Average qualitative scores obtained by a group of 12 people comparing pairs of differently enlarged (4X) images and selecting for each pair the preferred one. The score of each algorithm is the average number of preferences of the algorithm on the 5 comparisons made by each person on each image.

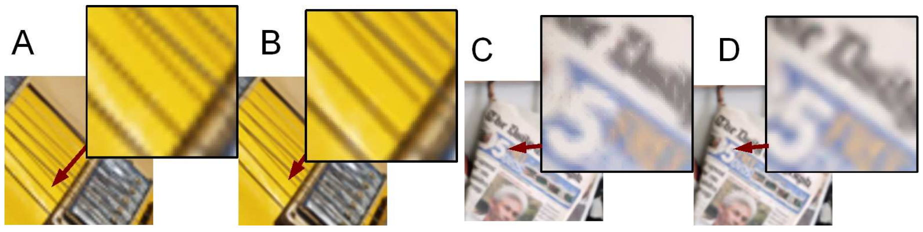

Figure 6: A: 4X enlargement obtained with Bicubic interpolation: jagged artifacts are evident. B: Same detail enlarged with ICBI. Artifacts are completely removed. C: 4X enlargement obtained with improved NEDI: directional artifacts are evident. D: Same detail enlarged with ICBI. Artifacts are completely removed.

5 Discussion

In this paper we presented two new general purpose image interpolation algorithms. The first (FCBI), like many others, puts original pixels in an enlarged grid and fill holes averaging neighboring pixels in two steps. The new idea is to interpolate directionally in order to preserve locally second order derivatives. The second method (ICBI) adds to the previous method an iterative refinement that try to reduce second order derivatives changes keeping original pixel values fixed and also preserving large gray level variations and smoothing isophote lines. This procedure creates images that have a natural aspect and present less artifacts than those obtained with other techniques in literature. Furthermore, the method is not computationally heavy as iNEDI or example based methods and is suitable for real time applications. Experimental results (qualitative and quantitative tests on heterogeneous natural images) show the advantages of the approach proposed. Matlab code with GPL license and a compiled C version of ICBI are available at the web site **www.andreagiachetti.it/icbi**. We plan to improve our method by using a multiresolution derivative estimation and to optimize implementations to reach real time performances.

Acknowledgements: Thanks to Xin Li for providing original NEDI implementation. Annalisa Brughitta checked the English.

References

[1] N. Asuni and A. Giachetti. Accuracy improvements and artifacts removal in edge based image interpolation. In *Proc. 3rd Int. Conf. Computer Vision Theory and Applications (VISAPP'08)*, 2008.

[2] C. B. Atkins, C. A. Bouman, and J. P. Allebach. Optimal image scaling using pixel classification. In *Proc. IEEE Int. Conf. Im. Proc.*, volume 3, pages 864–867, 2001.

[3] S. Battiato, G. Gallo, and F. Stanco. A locally-adaptive zooming algorithm for digital images. *Image and Vision Computing*, 20:805–812, 2002.

[4] Y. Cha and S. Kim. The error-amended sharp edge (ease) scheme for image zooming. *IEEE Transactions on Image Processing*, 16:1496–1505, 2007.

[5] M.J. Chen, C.H. Huang, and W.L. Lee. A fast edge-oriented algorithm for image interpolation. *Image and Vision Computing*, 23:791–798, 2005.

[6] R. Fattal. Image upsampling via imposed edge statistics. *ACM Transactions on Graphics*, 26(3):95, 2007.

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

[8] Kenji Kamimura, Norimichi Tsumura, Toshiya Nakaguchi, Yoichi Miyake, and Hideto Motomura. Video super-resolution using texton substitution. In *SIGGRAPH '07: ACM SIGGRAPH 2007 posters*, page 63, New York, NY, USA, 2007. ACM.

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

[10] X. Lu, P.S. Hong, and M.J.T. Smith. An efficient directional image interpolation method. In *Proc. IEEE Int. Conf. Ac. Sp. Sign. Proc.*, volume 3.

[11] B.S. Morse and D. Schwartzwald. Image magnification using level-set reconstruction. In *Proc. IEEE Conf. Comp. Vis. Patt. Rec.*, volume 3, pages 333–340, 2001.

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

[13] R. R. Schultz and R. L. Stevenson. A bayesian approach to image expansion for improved definition. *IEEE Trans. Image Processing*, 3:233–242, 1994.

[14] D. Su and P. Willis. Image interpolation by pixel level data-dependent triangulation. *Computer Graphics Forum*, 23, 2004.

[15] J. Sun, N.N. Zheng, H. Tao, and H.Y. Shum. Image hallucination with primal sketch priors. In *Proceedings IEEE conf. on Comp. Vis. and Pat. Rec.*, volume 2, 2003.

[16] Y. Takahashi and A. Taguchi. An enlargement method of digital images with the prediction of high-frequency components. In *Proc. IEEE Int. Conf. Ac. Speech Signal Proc.*, volume 4, pages 3700–3703, 2002.

[17] Q. Wang and R. K. Ward. A new orientation-adaptive interpolation method. *IEEE Transactions on Image Processing*, 16:889–900, 2007.