EDGE-ADAPTIVE IMAGE INTERPOLATION USING CONSTRAINED LEAST SQUARES

Kazu Mishiba, Taizo Suzuki and Masaaki Ikehara

Department of Electronics and Electrical Engineering, Keio University
Yokohama, Kanagawa 223-8522, Japan

ABSTRACT

Some adaptive image interpolation methods have been proposed to create higher visual quality images than traditional interpolation methods such as bicubic interpolation. These methods, however, often suffer from high computational costs and unnatural texture interpolation. This paper proposes a novel edge-adaptive image interpolation method using an edge-directed smoothness filter. Our approach estimates the enlarged image from the original image based on an observation model. The estimated image is constrained to have many edge-directed smooth pixels which are measured by using the edge-directed smoothness filter introduced in this paper. Simulation results show that the proposal method produces images with higher visual quality, higher PSNRs and faster computational times than the conventional methods.

***Index Terms*—** image processing, adaptive image interpolation, edge-directed smoothness.

1. INTRODUCTION

Image interpolation is useful in many applications, for example, video upconversion is well-known as one of the most common ones. In recent years, conversion from SDTV signals to HDTV signals is especially performed with the spread of high definition devices. Several image interpolation methods have been developed because such upconversion requires higher quality image with faster implementation and lower cost. For example, there are convolution-based interpolation, polynomial interpolation and spline interpolation. The biggest advantage of these methods is low computational complexity. However, produced images using these methods sometimes suffer from visual artifacts such as unsmooth and blurred edge.

To improve these artifacts, adaptive interpolation methods have been proposed in recent years. They are classified into explicit and implicit methods. Implicit methods tend to create higher visual quality images compared with explicit methods. In [1], missing HR pixels are interpolated based on HR local covariance estimated from LR local covariance. To interpolate missing HR pixels to fit the local pixel structures, Zhang and Wu use a 2-D piecewise autoregressive model whose parameters are estimated in a moving window in the LR image [2]. Dong et al.[3] use the iterative back-projection technique which is regularized by a non-local regularizer to minimize the reconstruction error. Although these methods create high visual quality images, they require high computational cost and large memory to calculate. Also in some cases, textures which differ from ones in original images are produced.

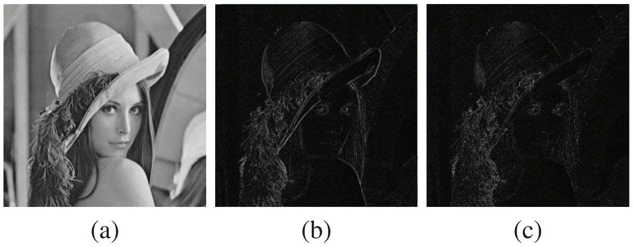

Fig. 1. Filtering results: (a)original image, (b)Laplacian filter, (c)edge-directed smoothness filter with $\sigma = 0.5$, $s = 3$ and $\beta = 0.01$.

This paper proposes a new edge-adaptive interpolation method to improve above problems. To estimate the HR image from its LR counterpart, our proposal method solves a constrained least squares problem based on an observation model and edge-directed smoothness measured by the edge-directed smoothness filter introduced in this paper. In addition, to obtain the HR image with faster implementation and lower cost, the problem is solved as a linear least squares problem.

2. EDGE-DIRECTED SMOOTHNESS FILTER

In this section, we introduce a new filter to quantify the degree of edge-directed smoothness in a local area. When the filter is applied in areas where are not smooth along edge directions, the filtering result has large energy. To obtain an interpolated image with edge-directed smoothness, we constrain the enlarged image applied the filter to have low energies.

Local smoothness in an image is evaluated by measuring high-frequency energy in the local area. Such energy measurement methods have been researched widely. As a representative example, Laplacian filter is simple and useful operator to measure it. Fig. 1(b) shows the energies of the image applied Laplacian filter to Fig. 1(a). As shown in Fig. 1(b), edge and texture areas have large high-frequency energies and smooth areas have small energies.

The second spatial derivative of intensity values in edge direction is one of the simplest measures of edge-directed smoothness. This is calculated by the difference between a pixel and the average of its neighboring pixels in edge direction. This calculation, however, is highly sensitive to noise. Additionally, it is not work well in an area with slight edge intensities, whose smoothness should be measured by calculating the difference between a pixel and the average of its neighboring pixels like Laplacian filter. We introduce a new edge-directed smoothness filter which solves the problem. The filter calculates the difference between a pixel and the weighted average of its neighboring pixels whose weights are determined by a 2D Gaussian function and vary with the condition of an edge. In an area with larger edge intensities, the weights in an edge direction become larger than in the direction perpendicular to the edge. In an area without edge, the weight distribution becomes isotropic like Laplacian filter. This weight control is achieved by changing a spread parameter σ of a Gaussian function in the direction perpendicular to the edge depending on edge intensities.

Let us denote the relation between a spatial filter c and the image d applied c to an image h by

$$d(\boldsymbol{u}) = \sum_{\boldsymbol{v} \in T_s} c_{(\boldsymbol{u},\boldsymbol{v})} h(\boldsymbol{u}+\boldsymbol{v}). \tag{1}$$

where T_s is a set of the spatial positions with their origin at the center of the filter size s, $\boldsymbol{u} = [u_x, u_y]^T$ and $\boldsymbol{v} = [v_x, v_y]^T$ indicate spatial positions in vector notation. In the case of $s = 3$, for example, $T_s = \{[1,1]^T, [1,0]^T, [1,-1]^T, [0,1]^T, [0,0]^T, [0,-1]^T, [-1,1]^T, [-1,0]^T, [-1,-1]^T\}$. The edge-directed smoothness filter $c_{(\boldsymbol{u},\boldsymbol{v})}$ is given as

$$c_{(\boldsymbol{u},\boldsymbol{v}) \in T_s} = \begin{cases} -1 & \boldsymbol{v} = [0,0]^T \\ \alpha_{(\boldsymbol{u},\boldsymbol{v})} & otherwise \end{cases} \tag{2}$$

where

$$\begin{aligned} &\alpha_{(\boldsymbol{u},\boldsymbol{v})} = \nu \exp(-\frac{\|\boldsymbol{p(u,v)}\|^2}{2\sigma^2}), \\ &\boldsymbol{p(u,v)} = \boldsymbol{G}_{(\boldsymbol{u})} \boldsymbol{R}_{(\boldsymbol{\theta}_{(\boldsymbol{u})})} \boldsymbol{v}, \\ &\boldsymbol{G}_{(\boldsymbol{u})} = \begin{bmatrix} 1 & 0 \\ 0 & E(g_{(\boldsymbol{u})}) \end{bmatrix}, \boldsymbol{R}_{\boldsymbol{\theta}} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}, \end{aligned} \tag{3}$$

ν is a normalization parameter satisfying $\sum c = 0$, σ is a spread parameter, $g_{(\boldsymbol{u})}$ and $\theta_{(\boldsymbol{u})}$ are an edge intensity and an edge direction at $\boldsymbol{u}$, respectively, and $E(g_{(\boldsymbol{u})})$ is a monotonically increasing function satisfying $E(0) \geq 1$. We use $E(g_{(\boldsymbol{u})}) = \beta g_{(\boldsymbol{u})} + 1$ where $\beta (> 0)$ is an edge intensity parameter. Scaling matrix $G_{(\boldsymbol{u})}$ makes pixel weights in an edge direction larger than ones in a direction perpendicular to the edge direction.

Here we show the calculation process of the coefficients of the edge-directed smoothness filter. First, the edge intensities and directions in local areas are calculated. These can be obtained by using the Sobel operator as follows. Let h'_x and h'_y denote the results obtained by applying the Sobel operator in the horizontal and vertical directions to an image h, respectively. The edge directions θ and intensities g are given by

$$\theta = \tan^{-1}\left(-\frac{h'_x}{h'_y}\right), g = \sqrt{{h'_x}^2 + {h'_y}^2}. \tag{4}$$

Next, substitution of (4) in (3) gives the coefficients of the edge-directed smoothness filter. Fig. 1(c) shows the result of applying the edge-directed smoothness filter to Fig. 1(a) with $\sigma = 0.5, \beta = 0.01$ and $s = 3$. The zoomed images of the original image and the applied image are shown in Fig. 2. The applied image has low energies in an edge-directed smooth area and high energies in a complex area.

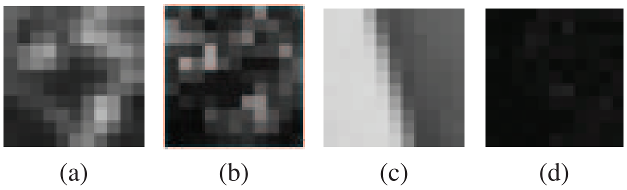
(a) (b) (c) (d)

Fig. 2. Zoomed images of part of the original image and the image applied the edge-directed smoothness filter. (a)A complex area in the original image. (b)The applied image of (a). (c)An edge-directed smooth area in the original image. (d)The applied image of (c).

3. INTERPOLATION ALGORITHM

In this section, we describe our new image interpolation algorithm using the edge-directed smoothness filter introduced in section 2. Our interpolation algorithm is based on an observation model, which formulates the relation between an HR image and an LR image. We use the following model: an LR image f of size $N_1 \times N_2$ results from warping, blurring, subsampled on an HR image h of size $L_1 N_1 \times L_2 N_2$, and added noise [4]. L_1 and L_2 represent subsampling factors in the horizontal and vertical directions, respectively. From the view point of image interpolation, L_1 and L_2 are considered as enlargement factors. For simplicity, we consider the case that L_1 and L_2 are integers satisfying $L_1 = L_2 = L \geq 2$. Let $\boldsymbol{f}$ and $\boldsymbol{h}$ denote lexicographically ordered vectors of size $N_1 N_2 \times 1$ and $L^2 N_1 N_2 \times 1$, respectively. Here $\boldsymbol{f}$ and $\boldsymbol{h}$ are composed of pixel values of the image f and h, respectively. The relation between $\boldsymbol{f}$ and $\boldsymbol{h}$ can be expressed as

$$\boldsymbol{f} = \boldsymbol{W}\boldsymbol{h} + \boldsymbol{n} \tag{5}$$

where $\boldsymbol{W}$ is a warping, blurring and subsampling matrix of size $N_1 N_2 \times L^2 N_1 N_2$ and $\boldsymbol{n}$ is a lexicographically ordered noise vector.

In image interpolation, only subsampling and blurring caused by the point spread function (PSF) of the sensor are considered since there is no need to consider a camera motion between the HR image and the LR image. This blurring and subsampling are accomplished by averaging a square block of HR pixels. Here let B_{f_i} denote a set of HR pixels which are contained in the HR square block corresponding to the position of the LR pixel f_i. The relation between LR pixels $f(x, y)$ and HR pixels $h(x, y)$ is represented by

$$f(\boldsymbol{u}) = \frac{1}{L^2} \sum_{\boldsymbol{v} \in T'_L} h(L\boldsymbol{u} + \boldsymbol{v}). \tag{6}$$

where $\boldsymbol{v} \in T'_L$ is a set of the spatial positions satisfying $0 \leq v_x, v_y < L$. From (5) and (6), the interpolated image h is expressed in matrix notation as

$$\hat{\boldsymbol{h}} = \arg\min_{\boldsymbol{h}} \|\boldsymbol{f} - \boldsymbol{W}\boldsymbol{h}\|^2 \tag{7}$$

where entry $w_{i,j}$ of $\boldsymbol{W}$ satisfy $w_{i,j} = 1/L^2$ if $h_j \in B_{f_i}$ and otherwise $w_{i,j} = 0$. However, the problem of solving (7) is an ill-posed. To make the problem well-posed, (7) is regularized by using the prior knowledge which is represented by the edge-directed smoothness constraint. We regularize (7) as

$$\hat{\boldsymbol{h}} = \arg\min_{\boldsymbol{h}} \|\boldsymbol{f} - \boldsymbol{W}\boldsymbol{h}\|^2 + \lambda \|\boldsymbol{C}\boldsymbol{h}\|^2 \tag{8}$$

where λ is a regularization parameter and entry $c_{i,j}$ of $\boldsymbol{C}$ satisfy $c_{i,j} = c_{(\boldsymbol{u},\boldsymbol{v})}$ if $\boldsymbol{u} \in T_s$ and otherwise $c_{i,j} = 0$, where $h(\boldsymbol{u})$ equals to the position h_i and $h(\boldsymbol{u} + \boldsymbol{v})$ to h_j. This regularization is to obtain an interpolated image which has many edge-directed smooth pixels. In (8), λ controls the balance between the fidelity to the LR image intensity and the smoothness in edge directions.

The HR edge information to calculate C in (8) is derived not from an estimated HR image h but from the LR image f. There are two reasons to do so. One is to preserve the LR edge structure in the HR image. Another is to solve (8) as a linear least squares problem. The process of obtaining the HR edge information is as follows. First, the LR horizontal edge f'_x and vertical edge f'_y are obtained by applying the Sobel operator to the original image f. Next, the HR horizontal edge h'_x and the vertical edge h'_y are estimated by interpolating f'_x and f'_y to the HR image size by using bilinear interpolation method. Last, substitution of h'_x and h'_y in (4) gives the estimated HR edge information. Using the obtained C, (8) can be rewritten as the linear least squares problem

$$\boldsymbol{A}\boldsymbol{h} \approx \boldsymbol{b} \tag{9}$$

where

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{W} \\ \sqrt{\lambda}\boldsymbol{C} \end{bmatrix}, \ \boldsymbol{b} = \begin{bmatrix} \boldsymbol{f}^T 0 \cdots 0 \end{bmatrix}^T.$$

Solving (9) gives an interpolated image.

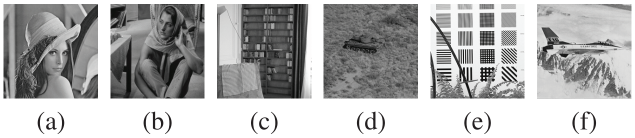

Fig. 3. Test images: (a)Lena, (b)Barbara, (c)Room, (d)Tank, (e)Chart, (f)Airplane.

Table 1. PSNRs[dB] and average computation times[sec].

Name	Bicubic	NEDI[1]	SAI[2]	NLBP[3]	Proposal
Lena	34.07	33.63	34.65	31.46	**35.46**
Barbara	25.40	24.67	25.34	24.51	**25.66**
Room	28.40	29.04	29.78	26.80	**30.92**
Tank	31.18	30.48	31.12	29.46	**31.80**
Chart	18.28	18.05	17.38	**19.44**	19.34
Airplane	31.26	29.64	31.13	28.88	**32.72**
time[sec]	0.012	20.28	*2.44	*5.74	2.03

*Computation times of SAI and NLBP are obtained with executable files. The others are implemented with Matlab.

4. SIMULATION RESULTS

In this section, we compare the proposal method with bicubic method, NEDI method [1], SAI method [2] and NLBP method [3] to validate our proposal algorithm. Fig. 3 lists six example images in our test set. We obtained simulated LR images by using the function *imresize* in MATLAB Image Processing Toolbox with a downsampling factor of 2. All algorithms run using default parameters. In our method, we found that $\sigma = 0.5, \beta = 0.001$, $s = 3$ and $\lambda = 0.1$ produce sufficiently good results. We use these values in all our experimental results. To solve the linear least squares problem (9), we use the conjugate gradient method. All the experiments were taken on an Intel Core i7 3.2GHz with 6GB of memory.

The PSNR and the average computational time for each interpolation algorithm are shown in Table 1. The PSNR results of the proposal method show better performance than other interpolation methods for most test images. The proposal method has the advantage of computational speed over the other adaptive interpolation methods. The bicubic interpolation method is faster than other methods. However, the produced images have unsmooth and blurred edges more than other methods as shown in Fig. 4(b). The NEDI method creates steep edges but suffers from noisy interpolation artifact in areas which have multiple edges as shown in Fig. 4(c) and Fig. 5(c). Additionally, texture regions are smoothed and these details are lost as shown in Fig. 6(c). The SAI method creates the smoothest edges in all methods. However, this method produces inadequate pixel structures which do not exist in original images as shown in Fig. 4-6. In Fig. 4(d), some unconnected stripes in the original image are connected due to over-smoothing. Loss of details can be seen in Fig. 6(d). The NLBP method over-enhances edges as shown in Fig. 4(e) and Fig. 6(e). This leads to some artifacts and results in lower PSNRs. Only in Chart image, which has sharp edges and many flat regions, the method produces an image

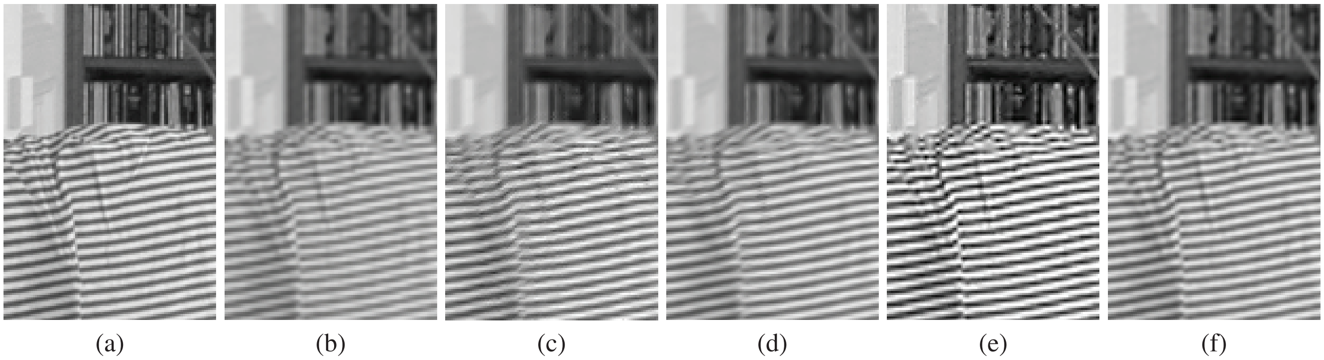

Fig. 4. Comparison on Room image: (a)original image, (b)bicubic, (c)NEDI, (d)SAI, (e)NLBP, (f)proposal method.

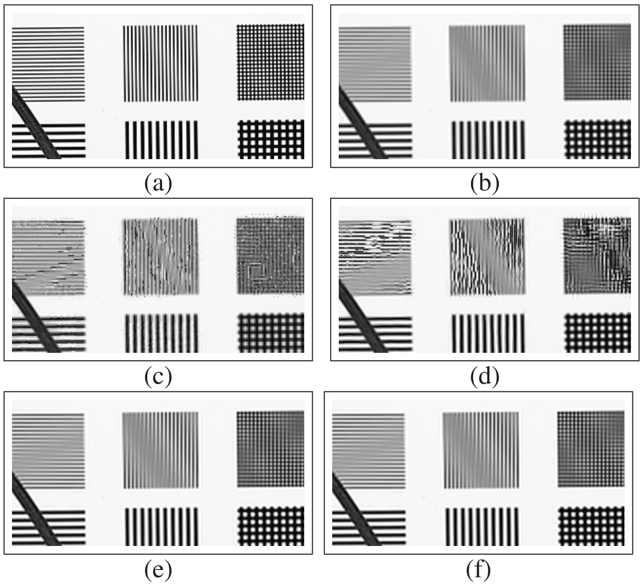

Fig. 5. Comparison on Chart image: (a)original image, (b)bicubic, (c)NEDI, (d)SAI, (e)NLBP, (f)proposal method.

with the highest PSNR. The proposal method produces sharp edges and edge-directed smoothness pixels as shown in Fig. 4(f). In Fig. 5(f), production of inadequate interpolated pixels is improved compared with other adaptive interpolation methods. Keeping details can be seen in Fig. 6(f).

5. CONCLUSION

We propose an adaptive image interpolation method using the edge-directed smoothness filter. Our method constrains the interpolated image to have edge-directed smoothness and the fidelity to the original image data based on the observation model. Thereby our proposal method produces images with high visual quality and improves some artifacts. Furthermore, the proposed method also performs well on PSNRs and computational times compared with other adaptive interpolation methods.

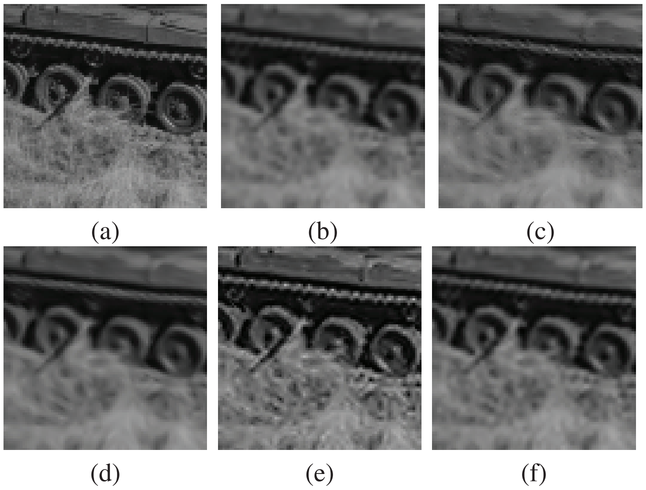

Fig. 6. Comparison on Tank image: (a)original image, (b)bicubic, (c)NEDI, (d)SAI, (e)NLBP, (f)proposal method.

Acknowledgement This work was supported by Global COE Program ”High-Level Global Cooperation for Leading-Edge Platform on Access Spaces (C12)”.

6. REFERENCES

[1] Xin Li and Michael T. Orchard, “New edge-directed interpolation,” *IEEE Transactions on Image Processing*, vol. 10, pp. 1521–1527, 2001.

[2] Xiangjun Zhang and Xiaolin Wu, “Image interpolation by adaptive 2-d autoregressive modeling and soft-decision estimation,” *Image Processing, IEEE Transactions on*, vol. 17, no. 6, pp. 887–896, June 2008.

[3] W.S. Dong, L. Zhang, G.M. Shi, and X.L. Wu, “Nonlocal back-projection for adaptive image enlargement,” in *ICIP 2009*, 2009, pp. 349–352.

[4] Sung C. Park, Min K. Park, and Moon G. Kang, “Super-resolution image reconstruction: a technical overview,” *Signal Processing Magazine, IEEE*, vol. 20, no. 3, pp. 21–36, 2003.