PAPER

Image Interpolation Using Edge-Directed Smoothness Measure Filter

Kazu MISHIBA^{†a)}, *Student Member* and Masaaki IKEHARA[†], *Member*

SUMMARY This paper proposes a novel adaptive image interpolation method using an edge-directed smoothness filter. Adaptive image interpolation methods tend to create higher visual quality images than traditional interpolation methods such as bicubic interpolation. These methods, however, often suffer from high computational costs and production of inadequate interpolated pixels. We propose a novel method to overcome these problems. Our approach is to estimate the enlarged image from the original image based on an observation model. Obtaining an image with edge-directed smoothness, we constrain the estimated image to have many edge-directed smooth pixels which are measured by using the edge-directed smoothness filter introduced in this paper. Additionally, we also propose a simplification of our algorithm to run with lower computational complexity and smaller memory. Simulation results show that the proposal method produces images with high visual quality and performs well on PSNR and computational times.

key words: *adaptive image interpolation, edge-directed smoothness, observation model*

1. Introduction

Image interpolation is useful in many applications. A commonly used application is production of high resolution (HR) images with low resolution (LR) image sensors. Although the resolutions of image sensors have been increased, using image interpolation to obtain HR images has the advantage of low costs and being able to use LR imaging systems. Another application is video upconversion. In recent years, conversion from SDTV signals to HDTV signals is especially performed with the spread of high definition devices. Faster upconversion methods creating higher quality images at a lower cost have been required.

Several image interpolation methods have been developed. Widely used interpolation methods are convolution-based interpolation, polynomial interpolation and spline interpolation [1]. The biggest advantage of these methods is low computational complexity. However, produced images using these methods suffer from visual artifacts such as unsmooth and blurred edge.

To improve these artifacts, adaptive interpolation methods have been proposed in recent years. Adaptive interpolation methods are classified into two types. One class is explicit adaptive interpolation methods. In [2], HR edge map is generated and then HR pixels are interpolated with bi-

linear interpolation modified to prevent interpolation across the edges. In [3], subpixel edges are localized and then HR pixels are interpolated to fit the continuous step edges. Another class is implicit adaptive interpolation methods. In [4], 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 [5].

Implicit adaptive interpolation methods tend to create higher visual quality compared with explicit methods. However, there are still some disadvantages. These methods require high computational cost and large memory to calculate. Also in some cases, textures which differ from original images are produced. Especially, it is desired to improve first disadvantage for consumer embedded devices, whose available resources are often limited.

We propose a new implicit adaptive interpolation method to improve the above problems. Our interpolation approach is to estimate the HR image from its LR counterpart based on an observation model. The estimated HR image is constrained to have many edge-directed smooth pixels which are measured by using the edge-directed smoothness filter introduced in this paper. To obtain the HR image, we solve a constrained least squares problem using the edge-directed smoothness regularizer. Unlike other deterministic approaches using a regularizer with edge-preserving smoothing, such as bilateral filtering and total variation filtering, our approach attempts to minimize edge-directed smoothness itself. Therefore, our method is expected to create smoother pixels in edge directions compared with the other approaches. Additionally, our method has the advantage of the computational speed. We also propose a simplification of our algorithm to run with lower computational complexity and smaller memory.

The rest of this paper is organized as follows. The next section introduces the edge-directed smoothness measure. Our new interpolation algorithm using the measure is described in Sect. 3. Section 4 shows a simplification of the algorithm for faster computation and smaller memory requirement. We discuss the results in Sect. 5 and conclude in Sect. 6.

2. Edge-Directed Smoothness Measure Filter

In this section, we introduce a new filter to quantify the degree of edge-directed smoothness in a local area. In our in-

Manuscript received October 14, 2009.

Manuscript revised January 21, 2010.

[†]The authors are with the Department of Electronics and Electrical Engineering, Keio University, Yokohama-shi, 223-8522 Japan.

a) E-mail: mishiba@tkhm.elec.keio.ac.jp

DOI: 10.1587/transinf.E93.D.1618

terpolation method, this filter is used to improve unsmooth edge, which often occurs with bicubic interpolation. In areas where are not smooth in edge direction, the result of applying this filter has large energy. In our method, obtaining an interpolated image with edge-directed smoothness, we constrain the enlarged image applied the filter to have low energies.

One of the methods to measure local smoothness in an image is to measure high-frequency energy in the local area. Small high-frequency energy means a smoothness area. Various methods to measure high-frequency energy in images have been presented. Laplacian is simple and useful operator to measure it. Laplacian is a 2D isotropic measure of the second spatial derivative. A spatially-discretized Laplacian filter calculates the difference between a pixel and the average of its neighboring pixels. Figure 1 (b) shows the energies of the image applied the 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.

One of the simplest methods to measure edge-directed smoothness is calculating the second spatial derivative in edge direction. This is achieved by calculating the difference between a pixel and the average of its neighboring pixels in edge direction. This method, however, has two problems. One is highly sensitive to noise. The other is the average calculation of neighboring pixels only in edge direction. In an area having slight edge intensities, smoothness should be measured by calculating the difference between a pixel and the average of its neighboring pixels like the Laplacian filter. We introduce a new edge-directed smoothness filter, which solve these problems. This filter has two features. One is robustness against noise. This filter calculates the difference between a pixel and the weighted average of its neighboring pixels which weights are based on a 2D Gaussian function. The other is adaptive weights depending on edge intensities. In an area having larger edge intensities, the weights in an edge direction become larger than in the direction perpendicular to the edge. In an area having no edge, the weight distribution becomes isotropic like the Laplacian filter. This 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

Fig. 1 Results of applying filters. (a) Original image. (b) Laplacian filter. (c) Edge-directed smoothness filter with $\sigma = 0.5$, $s = 3$ and $\beta = 0.01$.

$$d(\mathbf{u}) = \sum_{\mathbf{v} \in T_s} c(\mathbf{u}, \mathbf{v}) h(\mathbf{u} + \mathbf{v}). \quad (1)$$

where T_s is a set of the spatial positions with their origin at the center of the filter window of size s , $\mathbf{u} = [u_x, u_y]^T$ and $\mathbf{v} = [v_x, v_y]^T$ indicate spatial positions in vector notation. In the case of $s = 3$, for example, $T_s = \{[1, 1]^T, [1, 0]^T, [1, -1]^T, [0, 1]^T, [0, 0]^T, [0, -1]^T, [-1, 1]^T, [-1, 0]^T, [-1, -1]^T\}$. Figure 2 shows an example of $\mathbf{u}$ and $\mathbf{v}$. The edge-directed smoothness filter $c(\mathbf{u}, \mathbf{v})$ is given as

$$c(\mathbf{u}, \mathbf{v}) \in T_s = \begin{cases} -1 & \mathbf{v} = [0, 0]^T \\ \alpha_{(\mathbf{u}, \mathbf{v})} & \text{otherwise} \end{cases} \quad (2)$$

where

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

ν is a normalization parameter satisfying $\sum c = 0$, σ is a spread parameter, $g_{(\mathbf{u})}$ and $\theta_{(\mathbf{u})}$ are an edge intensity and an edge direction at $\mathbf{u}$, respectively, and $E(g_{(\mathbf{u})})$ is a monotonically increasing function satisfying $E(0) \geq 1$. We use $E(g_{(\mathbf{u})}) = \beta g_{(\mathbf{u})} + 1$ where $\beta (> 0)$ is an edge intensity parameter. Scaling matrix $\mathbf{G}_{(\mathbf{u})}$ makes pixel weights in an edge direction larger than ones in a direction perpendicular to the edge direction. The derivation of (3) is given in Appendix. Figure 3 illustrates the visually-presented function (3) with two different edge intensities. As shown Fig. 3, the distribution is isotropic in a smooth area, and anisotropic in an edge

Fig. 2 An example of $\mathbf{u}$ and $\mathbf{v}$. $\mathbf{e}$ and θ indicate an edge vector and edge direction at $\mathbf{u}$, respectively. $\|\mathbf{e}\|$ gives edge intensity g .

Fig. 3 Visualization of $\alpha_{(\mathbf{u}, \mathbf{v})}$ in (3) with $\sigma = 0.5$ and $\theta = 45^\circ$. (a) $E(g_{(\mathbf{u})}) = 1$. (b) $E(g_{(\mathbf{u})}) = 2$.

Fig. 4 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).

area.

Here we provide an example of applying the edge-directed smoothness filter to an image. 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

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

Substitution of (4) in (3) gives the coefficients of the edge-directed smoothness filter. Figure 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. 4. The applied image has low energies in an edge-directed smooth area and high energies in a complex area.

Since typical natural images tend to have many flat regions and small intensity changes in edge directions, the images applied the edge-directed smoothness filter have a few regions where have high energies. In our interpolation method, obtaining an interpolated image with edge-directed smoothness, we constrain the enlarged image applied the filter to have low energies.

3. Interpolation Algorithm

In this section, we describe our new image interpolation algorithm using the edge-directed smoothness filter introduced in Sect. 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 [6], [7]. 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 $L_1 = L_2 = L \geq 2$. Using f of size $N_1 N_2 \times 1$ and h of size $L^2 N_1 N_2 \times 1$ which are lexicographically ordered vectors composed pixel values of the image f and h , respectively, the relation between f and h can be expressed in matrix notation as

$$f = Wh + n \quad (5)$$

where W is a warping, blurring and subsampling matrix of size $N_1 N_2 \times L^2 N_1 N_2$ and 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 sensor PSF is typically modeled as a spatial averaging operator. Hence 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

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

where $v \in T'_L$ 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

$$\hat{h} = \arg \min_h \|f - Wh\|^2 \quad (7)$$

where entry $w_{i,j}$ of 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

$$\hat{h} = \arg \min_h \|f - Wh\|^2 + \lambda \|Ch\|^2 \quad (8)$$

where λ is a regularization parameter and entry $c_{i,j}$ of C satisfy $c_{i,j} = c_{(u,v)}$ if $u \in T_s$ and otherwise $c_{i,j} = 0$, where $h(u)$ equals to the position h_i and $h(u + v)$ 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. The other 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

$$Ah \approx b \quad (9)$$

where

$$A = \begin{bmatrix} \mathbf{W} \\ \sqrt{\lambda} \mathbf{C} \end{bmatrix}, \mathbf{b} = \begin{bmatrix} f \\ 0 \\ \vdots \\ 0 \end{bmatrix}.$$

There are many methods of solving linear least squares problems. Since the matrix A in (9) is a sparse matrix, the conjugate gradient method is suitable for solving (9). To apply the conjugate gradient method to the linear least squares problem (9) which has a nonsymmetric and indefinite coefficient matrix, one can rewrite (9) as

$$A^T A \mathbf{h} = A^T \mathbf{b}. \quad (10)$$

Deterministic approaches like (8) are often used in image enlargement, in which Laplacian regularization is often used. The regularization is based on the assumption that most images are naturally smooth with limited high-frequency activity [7]. However, since Laplacian regularizer works as a smoothing operator, this approach makes sharp edges blurred. Meanwhile, the regularizer using edge-directed smoothness filter works as a smoothing operator in smooth areas and an edge-directed smoothing operator in edge areas. This prevents edges from blurring and makes edges smooth along edge directions.

4. Algorithm Simplification

In this section, we describe a simplification of our interpolation algorithm. The simplified algorithm operates with lower computational cost, smaller memory request and slight increase of visual degradation. The simplified processes are calculating each $c_{(u,v)}$ and solving (8).

Calculating each $c_{(u,v)}$ requires high computational cost. This is simplified by quantization of edge information and using a lookup table. The calculation process 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, using thresholds of edge intensities $Th_e = \{th_1, \dots, th_n\}$, f'_x and f'_y are quantized to a value $q = (th_m + th_{m+1})/2$ where $th_m \leq f' < th_{m+1}$ and f' indicates an edge intensity in horizontal or vertical direction. The look up table gives precalculated values $c_{(u,v)}$ for corresponding quantized edge intensities. Last, the coefficients in h_i are given by loading the values from the look up table with the edge information of f_j satisfying $h_i \in B_{f_j}$. Figure 5 shows the edge intensities distribution of Fig. 1 (a) and an example of thresholds. Coarser quantization in larger edge intensities is suitable since natural images have few large edge intensities as shown in Fig. 5.

The other modified process is solving (8). Some algorithms such as the conjugate gradient method for the numerical solution of linear least squares problems have fast convergence but require large memories and large-scale circuits. For our simplification method, we use the gradient method, which has slow convergence but requires a relatively small memory and small-scale circuit. In the gradient

Fig. 5 Edge intensities distribution of Fig. 4 (a) in horizontal and vertical directions. The broken lines indicate thresholds in simplification method.

method, the sequence of iterations is given by

$$\mathbf{h}^{n+1} = \mathbf{h}^n + \gamma \{ \mathbf{W}(\mathbf{f} - \mathbf{W}\mathbf{h}^n) - \lambda \mathbf{C}^T \mathbf{C} \mathbf{h}^n \} \quad (11)$$

where γ is the convergence parameter. For faster convergence, the initial values $\mathbf{h}^0$ are set to the pixel values of the interpolated image from the LR image f with bicubic interpolation method.

5. Experimental Result

In this section, we compare the proposal method and the simplified proposal method with bicubic method, NEDI method [4], SAI method [5] to validate our proposal algorithm. Figure 6 lists eight example images in our test set. The process of our comparison experiment is as follows. First, the LR images are obtained by low-pass filtering and subsampling original images by a factor of two in both horizontal and vertical directions. Next, the LR images are interpolated by factor of two by using each interpolation algorithm, and then the HR images are obtained. Last, the HR images are compared with the original images. This approach is commonly used in the analysis of image interpolation algorithms [8]. Bicubic interpolated images are obtained with Matlab built-in functions. NEDI and SAI interpolated images are obtained with Matlab implementation and by running the executable file downloaded from each author's website, respectively. The images interpolated by using the proposal method and the simplified method are obtained with Matlab implementation. 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. For the iterative optimization in the simplified method, $Th_e = \{-1024, -512, -256, -128, -64, 64, 128, 256, 512, 1024\}$, $\gamma = 1$ and the number of iteration is set to 10. All the experiments were taken on an Intel Core i7 3.2 GHz with 6 GB of memory.

The PSNR and the average computational time for each interpolation algorithm are shown in Table 1. The PSNR result of the proposal method is better than other interpolation

Fig. 6 Test images. (a) Lena. (b) Barbara. (c) Room. (d) Tank. (e) Chart. (f) Airplane. (g) Building. (h) Boat.

Fig. 7 Comparison on Room image. (a) Original image. (b) Bicubic. (c) NEDI. (d) SAI. (e) Proposal method.

Table 1 PSNR and average computation times.

Name	Bicubic	NEDI [4]	SAI [5]	Proposal	simplify
Lena	34.07	33.63	34.65	35.46	35.10
Barbara	25.40	24.67	25.34	25.66	25.28
Room	28.40	29.04	29.78	30.92	30.29
Tank	31.18	30.48	31.12	31.80	31.65
Chart	18.28	18.05	17.38	19.34	19.08
Airplane	31.26	29.64	31.13	32.72	31.98
Building	25.98	25.36	25.91	26.75	26.53
Boat	29.94	29.48	30.30	31.00	30.70
time [sec]	0.012	20.28	*2.44	2.03	0.78

*Computation time of SAI is obtained with an executable file. Other methods are implemented with Matlab.

methods for all test images. The PSNR of the simplified method differs on an average of less than 0.34 dB from the PSNR of the proposal method and the computational speed is two times faster. The proposal method and the simplified method have the advantage of computational speed over the other adaptive interpolation methods. Having low computational complexity and easy of parallelization, the simplified method may have more advantage of computational speed with C implementation and circuit implementation.

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. 7 (b) and Fig. 8 (b).

The NEDI method creates steep edges but suffers from noisy interpolation artifact in areas which have multiple edges as shown in Fig. 7 (c) and Fig. 9 (b). Additionally, texture regions are smoothed and lose details that can be seen in Fig. 10 (b).

The SAI method creates most smooth edges in all

Fig. 8 Comparison on Airplane image. (a) Original image. (b) Bicubic. (c) SAI. (d) Proposal method.

methods. However, this method produces inadequate pixel structures which do not exist in original images as shown in Fig. 7–10. In Fig. 7 (d), some unconnected stripes in the original image are connected due to over-smoothing. Loss of details can be seen in Fig. 10 (c).

The proposal method produces sharp edges and edge-directed smoothness pixels as shown in Fig. 7 (e). In Fig. 9 (d), production of inadequate interpolated pixels is

Fig. 9 Comparison on Chart image. (a) Original image. (b) NEDI. (c) SAI. (d) Proposal method.

Fig. 10 Comparison on Tank image. (a) Original image. (b) NEDI. (c) SAI. (d) Proposal method.

Fig. 11 Comparison on Lena image. (a) Original image. (b) Simplified method. (c) Proposal method.

improved compared with other adaptive interpolation methods. Keeping details can be seen in Fig. 10(d) and improving blurring in Fig. 8(d). The simplified method creates high visual quality images similar to the proposal method as shown in Fig. 11.

Finally, We show the effectiveness of our regularizer using the edge-directed smoothness filter by comparing Laplacian regularizer. As shown in Fig. 12, our regularizer prevents edges from blurring and makes edges smooth along

Fig. 12 Comparison with Laplacian regularization. (a) Original image. (b) Result using Laplacian regularizer. (c) Result using proposal regularizer.

edge directions. Additionally, our method has an average PSNR 0.16 dB higher than the method using Laplacian regularizer.

6. Conclusion

We propose an adaptive image interpolation method using the edge-directed smoothness filter. In this method, the interpolated image is constrained by edge-directed smoothness and the fidelity to the original image data based on the observation model. Our proposal method produces images with high visual quality and improves some artifacts which are produced when one uses adaptive interpolation methods. Furthermore, the proposed method also performs well on PSNR and computational times compared with other adaptive interpolation methods. The simplified method also produces high visual quality images similar to the proposal method with faster speed.

References

- [1] E. Meijering, "A chronology of interpolation: From ancient astronomy to modern signal and image processing," *Proc. IEEE*, pp.319–342, 2002.
- [2] J. Allebach and P.W. Wong, "Edge-directed interpolation," *Proc. IEEE Int. Conf. Image Processing*, pp.707–710, Sept. 1996.
- [3] K. Jensen and D. Anastassiou, "Subpixel edge localization and the interpolation of still images," *IEEE Trans. Image Process.*, vol.4, no.3, pp.285–295, March 1995.
- [4] X. Li and M.T. Orchard, "New edge-directed interpolation," *IEEE Trans. Image Process.*, vol.10, pp.1521–1527, 2001.
- [5] X. Zhang and X. Wu, "Image interpolation by adaptive 2-d autoregressive modeling and soft-decision estimation," *IEEE Trans. Image Process.*, vol.17, no.6, pp.887–896, June 2008.
- [6] M. Elad and A. Feuer, "Restoration of a single superresolution image from several blurred, noisy, and undersampled measured images," *IEEE Trans. Image Process.*, vol.6, no.12, pp.1646–58, 1997.
- [7] S.C. Park, M.K. Park, and M.G. Kang, "Super-resolution image reconstruction: A technical overview," *IEEE Signal Process. Mag.*, vol.20, no.3, pp.21–36, 2003.
- [8] J.W. Hwang and H.S. Lee, "Adaptive image interpolation based on local gradient features," *IEEE Signal Processing Lett.*, vol.11, no.3, pp.359–362, 2004.

Appendix: Derivation of Coefficients (3)

The edge-directed smoothness filter calculates the difference between a pixel and the weighted average of its neighboring

pixels. We introduce the weighting coefficients based on the Gaussian function. Let us consider an orthogonal coordinate system whose axes (s, t) are rotated by angle θ with respect to x-y axes. Here s is an edge direction, t is the direction perpendicular to the edge and θ is an edge angle. In the edge direction s , the weighing distribution is given as

$$G(s, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{s^2}{2\sigma^2}\right), \quad (\text{A} \cdot 1)$$

where σ is a spread parameter. The weights in the direction t are required to be smaller depending on edge intensities than the weights in the direction s . This is achieved by changing the spread parameter depending on edge intensities. Let σ' denote the spread parameter in the direction t satisfying $\sigma' = \sigma/E(g)$ where $E(g)$ is a monotonically increasing function satisfying $E(0) \geq 1$ and g is an edge intensities. The weighing distribution in the direction t is given as

$$\begin{aligned} G(t, \sigma') &= \frac{1}{\sqrt{2\pi}\sigma'} \exp\left(-\frac{t^2}{2\sigma'^2}\right) \\ &= \frac{1}{\sqrt{2\pi}\sigma E(g)} \exp\left(-\frac{(tE(g))^2}{2\sigma^2}\right). \end{aligned} \quad (\text{A} \cdot 2)$$

By combining equations (A·1) and (A·2), we get

$$G(s, t, \sigma, g) = \frac{1}{2\pi\sigma^2 E(g)} \exp\left(-\frac{s^2 + (tE(g))^2}{2\sigma^2}\right). \quad (\text{A} \cdot 3)$$

Mapping a vector $\mathbf{v}$ from x-y space to s-t space is given by $\mathbf{R}\mathbf{v}$ where $\mathbf{R}$ is denoted by

$$\mathbf{R} = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix}. \quad (\text{A} \cdot 4)$$

Hence we get the equations as follows.

$$\begin{aligned} \begin{bmatrix} s \\ tE(g) \end{bmatrix} &= \begin{bmatrix} 1 & 0 \\ 0 & E(g) \end{bmatrix} \begin{bmatrix} s \\ t \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 \\ 0 & E(g) \end{bmatrix} \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \mathbf{v} \\ &= \mathbf{G}\mathbf{R}\mathbf{v}. \end{aligned} \quad (\text{A} \cdot 5)$$

Substitution of (A·5) in (A·3) and normalization lead to (3).

Kazu Mishiba received the B.E. and M.E. degrees in electrical engineering from Keio University, Yokohama, Japan, in 2004 and 2006, respectively. His research interests are in the areas of image interpolation and 3D remeshing.

Masaaki Ikehara received the B.E., M.E., and Dr.Eng. degrees in electrical engineering from Keio University, Yokohama, Japan, in 1984, 1986, and 1989, respectively. He was appointed Lecturer at Nagasaki University, Nagasaki, Japan, from 1989 to 1992. In 1992, he joined the Faculty of Engineering, Keio University. From 1996 to 1998, he was a Visiting Researcher at the University of Wisconsin, Madison, and Boston University, Boston, MA. He is currently a Professor with the Department of Electronics and Electrical Engineering, Keio University. His research interests are in the areas of multirate signal processing, wavelet image coding, and filter design problems.