Unconstrained Automatic Image Matching Using Multiresolutional Critical-Point Filters

Yoshihisa Shinagawa, *Member, IEEE Computer Society*, and Tosiyasu L. Kunii, *Fellow, IEEE*

Abstract—This paper proposes a novel method for matching images. The results can be used for a variety of applications: fully automatic morphing, object recognition, stereo photogrammetry, and volume rendering. Optimal mappings between the given images are computed automatically using multiresolutional nonlinear filters that extract the critical points of the images of each resolution. Parameters are set completely automatically by dynamical computation analogous to human visual systems. No prior knowledge about the objects is necessary. The matching results can be used to generate intermediate views when given two different views of objects. When used for morphing, our method automatically transforms the given images. There is no need for manually specifying the correspondence between the two images. When used for volume rendering, our method reconstructs the intermediate images between cross-sections accurately, even when the distance between them is long and the cross-sections vary widely in shape. A large number of experiments has been carried out to show the usefulness and capability of our method.

Index Terms—Image matching, multiresolution, nonlinear filters, critical-point filters, singularity, homotopy, image interpolation, morphing, volume rendering.

1 Introduction

Automatic matching of two images has been one of the most important and difficult themes of computer vision and computer graphics. For instance, once the images of an object from different view angles are matched, they can be used as the bases for generating other views [27], [11]. It has been, however, necessary to specify the corresponding points in the images manually. When the matching of right-eye and left-eye images is computed, it can be immediately used for stereo photogrammetry (e.g., [16], [15], [1], [5], [17], [6], [32], [28], [31]). Even if we reduce the number of candidate pairs of points taking advantage of epipolar lines, the complexity is high. To reduce the complexity, the coordinate values of a point in the left image are usually assumed to be close to those of the corresponding point in the right image. It has also been difficult to match global and local characteristics at the same time.

When a model facial image is matched with another facial image, it can be used to extract characteristic facial parts such as the eyes, nose and the mouth from the latter image. When two images of, for example, a man and a cat are matched exactly, all the in-between images can be generated and hence morphing can be done fully automatically, while in the existing methods, the correspondence of the points of the two images has to be specified manually [30], [4], [23], [25] and it is a tiresome work. In volume rendering (e.g., [20], [19]), a series of cross-sectional images are used for constituting voxels and a pixel in the upper cross-sectional image correspond to the pixel that occupies the same position in the lower cross section and this pair of pixels is used for the interpolation. Volume rendering suffers from unclear reconstruction of objects when the distance between the consecutive cross-sections is large and the cross-sections of the objects vary widely in shape. This is caused by the fact that the matching between the cross-sections is too simple (the pixels of the same image coordinate values are matched). Barequet and Sharir proposed a method of computing the rigid motion of two objects, each represented as a set of points in 3D space [3]. Their applications included the human organs. Their method, however, is not applicable to images.

This paper proposes a novel image matching method using a set of new multiresolutional filters called the *critical-point filters* to compute accurately the matching of images. There is no need for any prior knowledge of the objects. The matching of the images is computed at each resolution while going down from the coarse level to the fine level. Parameters are set completely automatically by dynamical computation analogous to human visual systems. Various applications such as the arbitrary view generation, morphing and volume rendering are shown to demonstrate the capability of our method.

A great number of image matching algorithms such as the stereo photogrammetry methods use the edge detection [16], [15], [1], [5], [17], [28]. The resulting matched pairs of points are sparse. To fill the gaps between the matched points, the disparity values are interpolated [13], [26], [6]. In general, all edge detectors suffer from the problem of judging whether a change in the pixel intensity in a local window they use really means the existence of an edge [8]. They suffer from noises because all edge detectors are high-pass filters by nature and hence detect noises at the same time. Our method does not need to detect edges.

- *Y. Shinagawa is with the Department of Information Science, The University of Tokyo, 7-3-1 Hongo, Bunkyo-Ku 113 Tokyo, Japan. E-mail: sinagawa@is.s.u-tokyo.ac.jp.*
- *T.L. Kunii is with the Computational Science Research Center, Hosei University.*

Manuscript received 21 Oct. 1996; revised 15 July 1998. Recommended for acceptance by D. Geiger.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number 107161.

0162-8828/98/$10.00 © 1998 IEEE

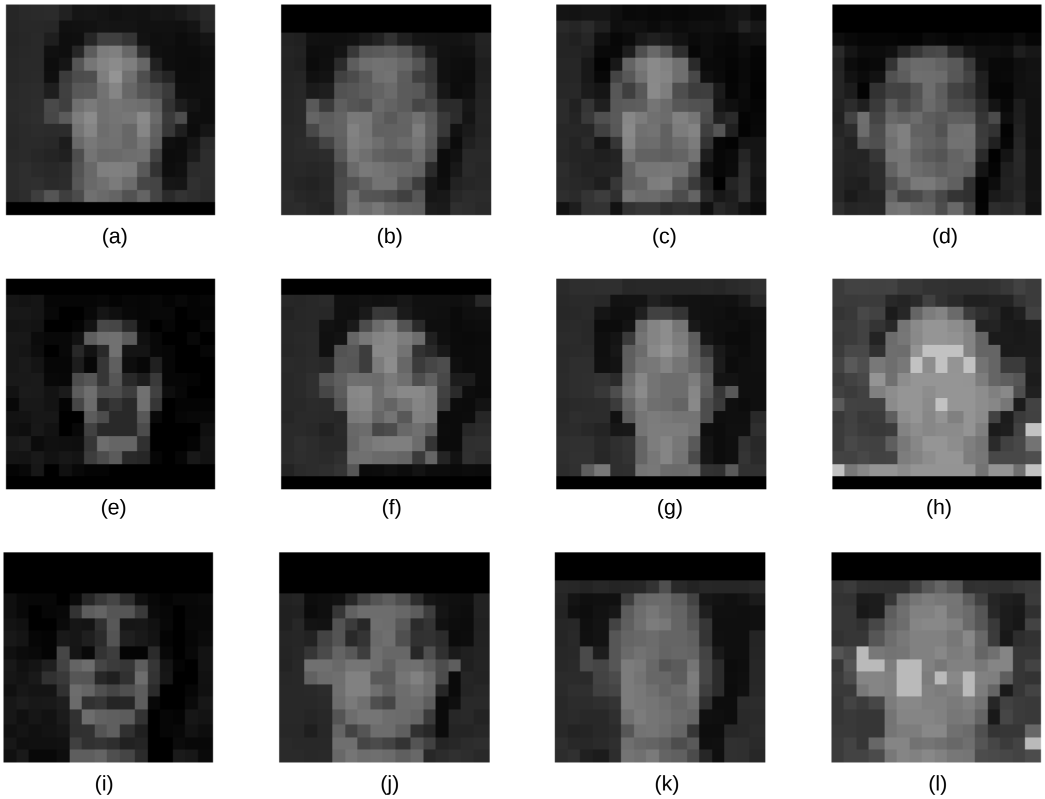


Fig. 1. The linear filters fail to match the characteristics of the images at the coarse levels of the resolution: (a) (b) the averaging filter and (c) (d) Battle-Lemarie wavelets. The critical-point filters (e) (i) $p^{(5,0)}$, (f) (j) $p^{(5,1)}$, (g) (k) $p^{(5,2)}$, and (h) (l) $p^{(5,3)}$, enable the correct matching of the same images.

2 THE HIERARCHY OF THE CRITICAL-POINT FILTERS

To recognize the global structures, a great number of multiresolutional filters have been proposed. They are classified into two groups: linear filters and nonlinear filters.

The linear filters have a long history and includes the Fourier transformations, Gaussian filter, and the wavelets (see, for example, [21], [22], [10], [9]). Such filters are equivalent to the convolutions of the image with some discretized kernels. The wavelets have become popular recently because they allow to both compose and decompose the images.

When used for image matching, however, the linear filters are not useful because the information of the pixel intensity of extrema as well as of their locations are blurred and become ambiguous. Figs. 1a and 1b, for example, show the results of the application of an averaging filter to the facial images in Figs. 7a and 7b, respectively. Figs. 1c and 1d show the results of the application of the scaling function of the Battle-Lemarié wavelet described in [21] to the same images. The information of the locations of the eyes (minima of the intensity) is ambiguous at this coarse level and hence it is impossible to compute the correct matching at this level of the resolution; i.e., the obtained global matching does not match the characteristics (eyes, i.e., the minima) correctly. Even when the eyes appear clearly at the finer level of resolution, it is too late to take back the errors introduced in the global matching. By smoothing the input images, stereo information in textured regions is also filtered out as pointed out in [31].

Recently, nonlinear filters has become available for morphological operations. For example, 1D sieve operators smooth out the images while preserving scale-space causality [2], [31]. A 1D sieve operator filters the original image by choosing the minimum (or the maximum) inside a window of a certain size. The resulting image is of the same size as the original one, but is simpler because small undulations are removed.

Our multiresolution filters preserve the intensity and the locations of each critical point of the images while reducing the resolution at the same time. Let the width of the image be N and the height be M. For simplicity, we assume in what follows the sizes of the images satisfy $N = M = 2^n$ where n is a positive integer. We denote an interval $[0, N] \subset \mathbf{R}$ by I. A pixel of the image at the location (i, j) is denoted by $p_{(i,j)}$ where $i, j \in I$. Let us construct a multiresolution hierarchy now, where the size of each image at the mth level is $2^m \times 2^m$ $(0 \le m \le n)$. The critical-point filter constructs the following four new images recursively:

$$p^{(m,0)}_{(i,j)} = \min\left(\min\left(p^{(m+1,0)}_{(2i,2j)}, p^{(m+1,0)}_{(2i,2j+1)}\right),\right.$$
$$\left.\min\left(p^{(m+1,0)}_{(2i+1,2j)}, p^{(m+1,0)}_{(2i+1,2j+1)}\right)\right)$$

$$p_{(i,j)}^{(m,1)} = \max\left(\min\left(p_{(2i,2j)}^{(m+1,1)}, p_{(2i,2j+1)}^{(m+1,1)}\right), \min\left(p_{(2i+1,2j)}^{(m+1,1)}, p_{(2i+1,2j+1)}^{(m+1,1)}\right)\right)$$

$$p_{(i,j)}^{(m,2)} = \min\left(\max\left(p_{(2i,2j)}^{(m+1,2)}, p_{(2i,2j+1)}^{(m+1,2)}\right), \max\left(p_{(2i+1,2j)}^{(m+1,2)}, p_{(2i+1,2j+1)}^{(m+1,2)}\right)\right)$$

$$p_{(i,j)}^{(m,3)} = \max\left(\max\left(p_{(2i,2j)}^{(m+1,3)}, p_{(2i,2j+1)}^{(m+1,3)}\right), \max\left(p_{(2i+1,2j)}^{(m+1,3)}, p_{(2i+1,2j+1)}^{(m+1,3)}\right)\right)$$

where

$$p_{(i,j)}^{(n,0)} = p_{(i,j)}^{(n,1)} = p_{(i,j)}^{(n,2)} = p_{(i,j)}^{(n,3)} = p_{(i,j)}.$$

Let us call the four images *subimages* in what follows. When we abbreviate $\min_{x \le t \le x+1}$ and $\max_{x \le t \le x+1}$ to α and β, respectively, the subimages can be abbreviated to $p^{(m,0)} = \alpha(x)\alpha(y)$ $p^{(m+1,0)}p^{(m,1)} = \alpha(x)\beta(y)p^{(m+1,1)}$, $p^{(m,2)} = \beta(x)\alpha(y)p^{(m+1,2)}$, and $p^{(m,3)}$ $= \beta(x)\beta(y)p^{(m+1,3)}$; i.e., they are analogous to the tensor products of α and β. From the singularity theoretical point of view, $\alpha(x)\alpha(y)$ preserves the maxima, $\beta(x)\beta(y)$ the minima, $\alpha(x)\beta(y)$ and $\beta(x)\alpha(y)$ the saddle points. This is the reason we call the filters *critical-point filters*. The minima are first matched by using $p^{(m,0)}$, the saddle points by $p^{(m,1)}$ based on the previous matching result regarding the maxima, other saddle points by $p^{(m,2)}$, and finally maxima by $p^{(m,3)}$.

Figs. 1e and 1i show the subimages $p^{(5,0)}$ of the same images in Figs. 1a and 1b, respectively, Figs. 1f and 1j $p^{(5,1)}$, Figs. 1g and 1k $p^{(5,2)}$, and Figs. 1h and 1l $p^{(5,3)}$, respectively. It is easy to see that the characteristics in the subimages can be matched clearly; i.e., the eyes can be matched by $p^{(5,0)}$, the mouths by $p^{(5,1)}$, the necks by $p^{(5,2)}$, and the ears by $p^{(5,3)}$.

3 COMPUTATION OF THE MAPPING BETWEEN IMAGES

First, let us consider two images. In this paper, the first one is called the **source image** and the second the **destination image** following the notations in [4]. The pixel of the source image at the location (i, j) is denoted by $p_{(i,j)}^{(n)}$ and that of the destination image at (k, l) by $q_{(k,l)}^{(n)}$, where i, j, k, l $\in I$. We then define the energy of the mapping between the images described in detail later. The energy is determined by the difference in the intensity of the pixel of the source image and its corresponding pixel of the destination image and by the smoothness of the mapping. First, the mapping $f^{(m,0)} : p^{(m,0)} \to q^{(m,0)}$ between $p^{(m,0)}$ and $q^{(m,0)}$ with the minimum energy is computed. Based on $f^{(m,0)}$, the mapping $f^{(m,1)}$ between $p^{(m,1)}$ and $q^{(m,1)}$ with the minimum energy is computed. The process continues until we finish computing $f^{(m,3)}$ between $p^{(m,3)}$ and $q^{(m,3)}$. Let us call each $f^{(m,i)}$ (i = 0, 1, 2, ...) a *submapping*. The order of i for computing $f^{(m,i)}$ can be changed as follows:

$$f^{(m,i)} : p^{(m,\sigma(i))} \to q^{(m,\sigma(i))}$$

where $\sigma(i) \in \{0, 1, .., 3\}$.

3.1 Bijectivity

The mappings we construct are the digital version of the *bijection*. A one-to-one surjective mapping is called a bijection. The bijectivity can be formalized as follows. In this paper, a pixel is identified with a grid point. The mapping of the source subimage to the destination subimage is represented by $f^{(m,s)} : I/2^{n-m} \times I/2^{n-m} \to I/2^{n-m} \times I/2^{n-m}$ (s = 0, 1, ..). $f^{(m,s)}(i, j) = (k, l)$ means that $p_{(i,j)}^{(m,s)}$ of the source image is mapped to $q_{(k,l)}^{(m,s)}$ of the destination image (s = 0, 1, ..). For simplicity, a pixel $q_{(k,l)}$ where $f(i, j) = (k, l)$ holds is denoted by $q_{f(i,j)}$ in this paper.

The definition of bijectivity is not trivial when the data sets are discrete as in our case of image pixels (grid points). We define the bijection in the case of discrete data sets as follows. (i, i', j, j', k, and l are integers in the following.)

First, we consider each square $p_{(i,j)}^{(m,s)} p_{(i+1,j)}^{(m,s)} p_{(i+1,j+1)}^{(m,s)} p_{(i,j+1)}^{(m,s)}$ on the source image plane denoted by R where $i = 0, ..., 2^m - 1$ and $j = 0, ..., 2^m - 1$. The edges of R are directed as $\overrightarrow{p_{(i,j)}^{(m,s)} p_{(i+1,j)}^{(m,s)}}$, $\overrightarrow{p_{(i+1,j)}^{(m,s)} p_{(i+1,j+1)}^{(m,s)}}$, $\overrightarrow{p_{(i+1,j+1)}^{(m,s)} p_{(i,j+1)}^{(m,s)}}$, and $\overrightarrow{p_{(i,j+1)}^{(m,s)} p_{(i,j)}^{(m,s)}}$. It is necessary that the square be mapped by f to a quadrilateral on the destination image plane. The quadrilateral $q_{f(i,j)}^{(m,s)} q_{f(i+1,j)}^{(m,s)} q_{f(i+1,j+1)}^{(m,s)} q_{f(i+1,j+1)}^{(m,s)}$ denoted by $f^{(m,s)}(R)$ should satisfy the following conditions (see Fig. 2).

3.1.1 Conditions (Bijectivity Conditions)

1) The edges of the quadrilateral $f^{(m,s)}(R)$ should not intersect one another.

2) The orientation of the edges of $f^{(m,s)}(R)$ should be the same as that of R (clockwise or counterclockwise).

3) The length of one edge of $f^{(m,s)}(R)$ can be zero to allow mappings that are retractions; i.e., $f^{(m,s)}(R)$ may be a triangle. It is not allowed, however, to be a point nor a line segment (figures of area 0).

The Bijectivity Conditions stated above is abbreviated **BC** in what follows.

In most cases, we further imposed the following condition to easily guarantee that the mapping is a surjection; we map each pixel on the boundary of the source image to the pixel that occupies the same location at the destination image; i.e.,

$$f(i, j) = (i, j) \text{ for } i = 0, i = 2^m - 1, j = 0, \text{ or } j = 2^m - 1.$$

This additional condition is abbreviated **SJ** in what follows.

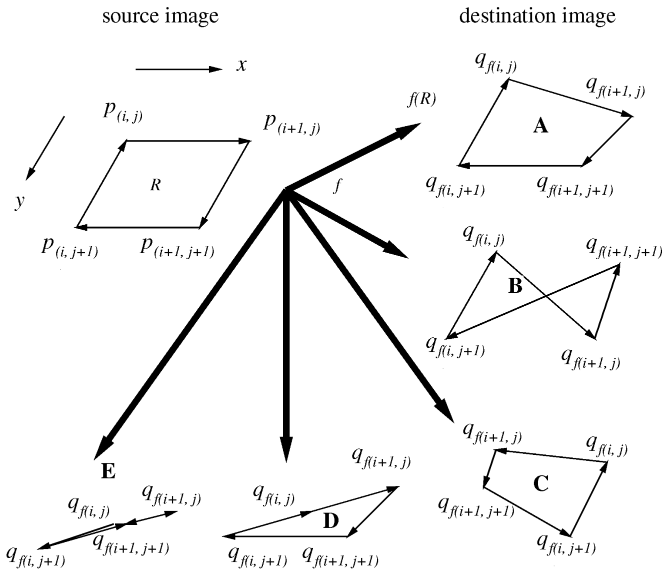


Fig. 2. The quadrilaterals *A* and *D* satisfy the Bijectivity Conditions and *B*, *C*, and *E* violate them.

3.2 The Energy of the Mapping

3.2.1 Cost Related to the Pixel Intensity

We then define the energy of the mapping *f*. We search for a mapping whose energy is minimum.

The energy is determined mainly by the difference in the intensity of the pixel of the source image and its corresponding pixel of the destination image. That is, the energy $C^{(m,s)}_{(i,j)}$ of the mapping $f^{(m,s)}$ at (i, j) is determined as

$$C^{(m,s)}_{(i,j)} = \left| V\left(p^{(m,s)}_{(i,j)}\right) - V\left(q^{(m,s)}_{f(i,j)}\right) \right|^2$$

where $V\left(p^{(m,s)}_{(i,j)}\right)$ and $V\left(q^{(m,s)}_{f(i,j)}\right)$ are the intensity values of the pixels $p^{(m,s)}_{(i,j)}$ and $q^{(m,s)}_{f(i,j)}$, respectively. In our experiments, the L_2 norm described above has been superior to the L_1 norm, enabling sharp matching.

The total energy $C^{(m,s)}_f$ of *f* is defined as the sum of $C^{(m,s)}_{(i,j)}$; i.e.,

$$C^{(m,s)}_f = \sum_{i=0}^{i=2^m-1} \sum_{j=0}^{j=2^m-1} C^{(m,s)}_{(i,j)}.$$

3.2.2 Cost Related to the Locations of the Pixel for Smooth Mapping

To obtain smooth mappings, we introduce another energy D_f of the mappings. It is determined by the locations of $p^{(m,s)}_{(i,j)}$ and $q^{(m,s)}_{f(i,j)}$ ($i = 0,.., 2^m - 1, j = 0, .., 2^m - 1$), regardless of the intensity of the pixels. The energy $D^{(m,s)}_{(i,j)}$ of the mapping $f^{(m,s)}$ at (i, j) is determined as

$$D^{(m,s)}_{(i,j)} = \eta E^{(m,s)}_{0(i,j)} + E^{(m,s)}_{1(i,j)}$$

where $\eta \geq 0$ is a real number and

$$E^{(m,s)}_{0(i,j)} = \left\| (i,j) - f^{(m,s)}(i,j) \right\|^2,$$

$$E^{(m,s)}_{1(i,j)} = \sum_{i'=i-1}^{i} \sum_{j'=j-1}^{j} \left\| \left(f^{(m,s)}(i,j) - (i,j)\right) - \left(f^{(m,s)}(i',j') - (i',j')\right) \right\|^2 \Big/ 4$$

where $\|(x,y)\| = \sqrt{x^2 + y^2}$. $f(i', j')$ is defined to be zero for $i' < 0$ or $j' < 0$.

E_0 is determined by the distance between (i, j) and $f(i, j)$. E_0 prevents a pixel being mapped to a pixel too far away from it. It is replaced by another energy function when the multiresolution approximation is introduced later. E_1 ensures the smoothness of the mapping. It represents the distance between the displacement of $p(i, j)$ and the displacements of its neighbors.

Finally, the energy D_f is determined by

$$D^{(m,s)}_f = \sum_{i=0}^{i=2^m-1} \sum_{j=0}^{j=2^m-1} D^{(m,s)}_{(i,j)}.$$

3.2.3 Total Energy of the Mapping

The total energy of the mapping is defined as $\lambda C^{(m,s)}_f + D^{(m,s)}_f$ where $\lambda \geq 0$ is a real number. λ can be regarded as the *temperature*. Our goal is to find a mapping that gives the minimum energy

$$\min_f \lambda C^{(m,s)}_f + D^{(m,s)}_f.$$

Note that the mapping becomes an identity mapping (i.e., $f^{(m,s)}(i, j) = (i, j)$ for all $i = 0, .., 2^m - 1$ and $j = 0, ..., 2^m - 1$) when $\lambda = 0$ and $\eta = 0$.

3.3 Determining the Mapping With Multiresolution

We look for a mapping $f_{\min}$ that gives the minimum energy and satisfies BC using the multiresolution hierarchy.

We compute the mappings between the source and destination images at each level of the resolution. Starting from the top of the resolution hierarchy, we determine the mapping at each level. The number of candidate mappings at each level is constrained by using the mapping at the upper level of the hierarchy.

$p^{(m-1,s)}_{(i',j')}$ (or $q^{(m-1,s)}_{(i',j')}$) is called the **parent** of $p^{(m,s)}_{(i,j)}$ (or $q^{(m,s)}_{(i,j)}$) when $(i', j') = \left(\left\lfloor \frac{i}{2} \right\rfloor, \left\lfloor \frac{j}{2} \right\rfloor\right)$ holds, respectively. Note that $\lfloor x \rfloor$ is the largest integer that does not exceed x. Conversely, $p^{(m,s)}_{(i,j)}$ (or $q^{(m,s)}_{(i,j)}$) is called the **child** of $p^{(m-1,s)}_{(i',j')}$ (or $q^{(m-1,s)}_{(i',j')}$), respectively. The function *parent*(i, j) is defined as

$$parent(i,j) = \left(\left\lfloor \tfrac{i}{2} \right\rfloor, \left\lfloor \tfrac{j}{2} \right\rfloor\right).$$

A mapping $f^{(m,s)}$ between $p^{(m,s)}_{(i,j)}$ and $q^{(m,s)}_{(k,l)}$ is then determined by computing its energy and finding the minimum one. The value $f^{(m,s)}(i, j) = (k, l)$ is determined as follows us-

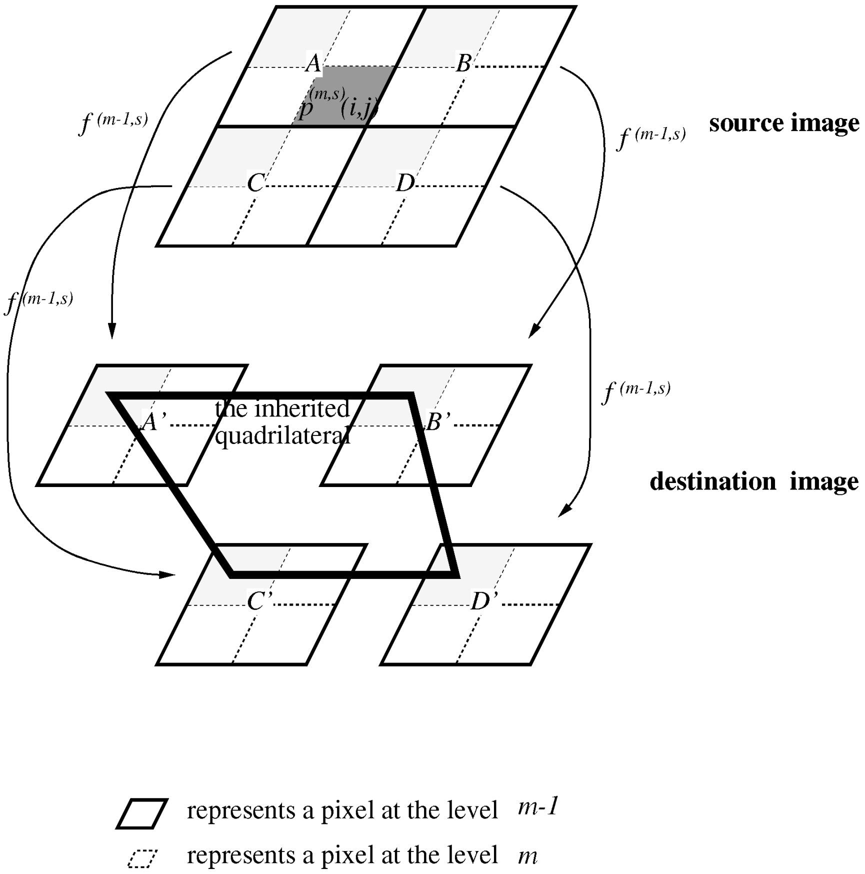

A
B
$p^{(m,s)}(i,j)$
C
D
f (m-1,s)
f (m-1,s)
f (m-1,s)
f (m-1,s)
source image
A'
the inherited quadrilateral
B'
destination image
C'
D'
represents a pixel at the level m-1
represents a pixel at the level m

Fig. 3. The inherited quadrilateral of $p^{(m,s)}_{(i,j)}$.

ing $f^{(m-1,s)}$ (m = 1, 2, ..., n).

First of all, $q^{(m,s)}_{(k,l)}$ should be inside a quadrilateral

$$q^{(m,s)}_{g^{(m,s)}(i-1,j-1)}\, q^{(m,s)}_{g^{(m,s)}(i-1,j+1)}\, q^{(m,s)}_{g^{(m,s)}(i+1,j+1)}\, q^{(m,s)}_{g^{(m,s)}(i+1,j-1)}$$

where

$$g^{(m,s)}(i, j) = f^{(m-1,s)}(parent(i, j)) + f^{(m-1,s)}(parent((i, j) + (1, 1))).$$

The quadrilateral defined above is referred to as the **inherited quadrilateral** of $p^{(m,s)}_{(i,j)}$ in what follows. Inside the inherited quadrilateral, we search for a pixel that minimizes the energy.

The above description is illustrated in Fig. 3 where the pixels A, B, C, and D of the source image are mapped to A', B', C', and D' of the destination image, respectively, at the level $m-1$ of the hierarchy. The pixel $p^{(m,s)}_{(i,j)}$ should be mapped to the pixel $q^{(m,s)}_{f^{(m)}(i,j)}$ that exists in the interior of the inherited quadrilateral $A'B'C'D'$.

The energy E_0 defined in the previous section is now replaced by

$$E_{0_{(i,j)}} = \left\| f^{(m,0)}(i,j) - g^{(m)}(i,j) \right\|^2$$

for computing the submapping $f^{(m,0)}$ and

$$E_{0_{(i,j)}} = \left\| f^{(m,s)}(i,j) - f^{(m,s-1)}(i,j) \right\|^2 (1 \le i)$$

for computing the submapping $f^{(m,s)}$ at the mth level. In this way, the mapping that keeps the energy of all the submappings low is obtained. The former equation represents the distance between $f(i, j)^{(m,s)}$ and the location where (i, j) should be mapped when regarded as a part of a pixel at the level $m - 1$.

When there is no pixel that satisfies the Bijectivity Conditions inside the inherited quadrilateral $A'B'C'D'$, we examine the pixels whose distance from the boundary of $A'B'C'D'$ is L (at first, L = 1). Among them, the one with the minimum energy satisfying the Bijectivity Conditions is chosen as the value of $f^{(m,s)}(i, j)$. L is increased until such a pixel is found or L reaches its upper bound $L^{(m)}_{\max}$. $L^{(m)}_{\max}$ is fixed for each level m. If we cannot find such a pixel at all, the third condition of the Bijectivity Conditions is abandoned temporarily for determining $f^{(m,s)}(i, j)$. If we still cannot find one, the first and the second conditions of the Bijectivity Conditions are abandoned next.

Multiresolution approximation is essential to determine the global correspondence of the images while avoiding the mapping being affected by small details of the images. Without the multiresolution approximation, it is impossible to detect correspondence between pixels whose distance is large and hence the size of an image is limited to be very small, and only tiny changes in the images can be handled as can be seen in [7]. Moreover, imposing smoothness on the mapping prevents finding the correspondence of such pixels because the energy of mapping a pixel to a pixel at a distance is high. The multiresolution approximation enables finding the appropriate correspondence of such pixels because the distance between them is small at the upper level of the hierarchy of the resolution.

4 Automatic Determination of the Optimal Parameter Values

One of the main deficiencies of existing image matching techniques lies in the difficulty of parameter tuning. In most cases, the tuning is done manually and it has been extremely difficult to choose the optimal values. Our method includes a theory for obtaining the optimal parameter values completely automatically.

Our system has two parameters: λ and η. In short, λ is the weight of the difference of the pixel intensity and η represents the stiffness of the mapping. These parameter values are increased step by step starting from zero. As λ gets larger, the value of $C_f^{(m,s)}$ for each submapping becomes smaller, which basically means that the two images are matched better. When the value exceeds the optimal value, however, the mapping becomes excessively distorted and $C_f^{(m,s)}$ begins to increase. We detect this threshold value for each submapping. This is analogous to the *focusing mechanism* of human visual systems where the images of the left and right eye are matched while moving one eye and the eye is fixed when the objects are clearly recognized (such a motion of the eye becomes noticeable when you see a random dot stereogram).

4.1 Dynamic Determination of λ

λ is varied from zero at a certain interval and each time a submapping is computed. Let us recall that the energy is defined by $\lambda C_f^{(m,s)} + D_f^{(m,s)}$. $D_f^{(m,s)}$ represents the smoothness and increases when the mapping becomes more distorted.

4.1.1 Normal Behavior of $C_{(i,j)}^{(m,s)}$ as a Function of λ

When we increase λ, a change in the mapping is not possible until $\lambda C_{(i,j)}^{(m,s)}$ becomes greater than one. It is because

$$D_{(i,j)}^{(m,s)} = \eta E_{0(i,j)}^{(m,s)} + E_{1(i,j)}^{(m,s)},$$

and $E_{0(i,j)}^{(m,s)}$ and $E_{1(i,j)}^{(m,s)}$ are positive integers that usually increase; i.e., the smallest step of change of $D_{(i,j)}^{(m,s)}$ is greater than one. If $\lambda C_{(i,j)}^{(m,s)}$ exceeds one, some pixels may move to a stabler state. Let us denote the number of such pixels by A. Let us also denote the histogram of $C_{(i,j)}^{(m,s)}$ by $h(l)$ where $h(l)$ is the number of pixels whose energy $C_{(i,j)}^{(m,s)}$ is l^2. In order that $\lambda l^2 \geq 1$ holds, we approximate l by $l^2 = \frac{1}{\lambda}$. When λ is slightly varied from λ_1 to λ_2,

$$A = \sum_{l=\left\lceil \frac{1}{\lambda_2} \right\rceil}^{\left\lfloor \frac{1}{\lambda_1} \right\rfloor} h(l) \simeq \int_{\frac{1}{\lambda_2}}^{\frac{1}{\lambda_1}} h(l)dl = -\int_{\lambda_2}^{\lambda_1} h(l)\frac{1}{\lambda^{3/2}}d\lambda = \int_{\lambda_1}^{\lambda_2}\frac{h(l)}{\lambda^{3/2}}d\lambda$$

pixels move to a stabler state. Let us assume that they move to a state where their energy is zero. Thus, the total energy decreases by $Al^2 = \frac{A}{\lambda}$. It means that the value of $C_f^{(m,s)}$ changes by $\partial C_f^{(m,s)} = -\frac{A}{\lambda}$. Therefore,

$$\frac{\partial C_f^{(m,s)}}{\partial \lambda} = -\frac{h(l)}{\lambda^{5/2}}$$

holds. As $h(l) > 0$, $C_f^{(m,s)}$ decreases in normal cases.

4.1.2 Excessive Distortion

When λ tries to go beyond the optimal value, however, the mapping becomes excessively distorted, and $D_f^{(m,s)}$ increases rapidly. In such a case, the total energy $\lambda C_f^{(m,s)} + D_f^{(m,s)}$ is dominated by $D_f^{(m,s)}$. The system tries to reduce the total energy by decreasing $D_f^{(m,s)}$ regardless of $C_f^{(m,s)}$. Thus, $C_f^{(m,s)}$ begins to increase. We detect this phenomenon to determine the optimal value of λ.

4.1.3 Typical Form of $C_f^{(m,s)}$

If we assume

$$h(l) = Hl^k = \frac{H}{\lambda^{k/2}}$$

where $H > 0$ and k are constants, we have $\frac{\partial C_f^{(m,s)}}{\partial \lambda} = -\frac{H}{\lambda^{5/2+k/2}}$. If $k \neq -3$, we obtain $C_f^{(m,s)} = \frac{H}{(3/2+k/2)\lambda^{3/2+k/2}}$, where C is a constant.

4.1.4 Foolproof for Preventing λ From Becoming Excessively Large

To make it sure, we also check the number of pixels violating BC to detect the optimal value of λ. Let us assume that the probability of violating BC when determining a mapping for each pixel is p_0. As

$$\frac{\partial A}{\partial \lambda} = \frac{h(l)}{\lambda^{3/2}}$$

holds, the number of pixels violating BC increases at the rate $B_0 = \frac{h(l)p_0}{\lambda^{3/2}}$ and, hence,

$$\frac{B_0 \lambda^{3/2}}{p_0 h(l)} = 1$$

is a constant. If we assume $h(l) = Hl^k$ as described before, for example,

$$B_0 \lambda^{3/2+k/2} = p_0 H$$

is a constant. When λ goes beyond the optimal value, however, the above value increases abruptly. We detect this phenomenon to determine the optimal value of λ by checking to see if $B_0\lambda^{3/2+k/2}/2^m$ exceeds an abnormal value $B_{0_{thres}}$. We also check the increasing rate B_1 of pixels violating the third condition of BC in the same way by checking to see if $B_1\lambda^{3/2+k/2}/2^m$ exceeds an abnormal value $B_{1_{thres}}$. The reason the factor 2^m is introduced is described later. The system is not sensitive to the two threshold values. They are used to detect abnormally excessive distortion of the mapping in case we fail to detect it by observing $C_f^{(m,s)}$ in Section 4.1.2.

4.1.5 The Upper Limit of λ

If λ exceeds 0.1 when computing the submapping $f^{(m,s)}$, we abandon the computation of $f^{(m,s)}$ and move on to the computation of $f^{(m,s+1)}$ using the previous submapping $f^{(m,s-1)}$ (s = 0, 1, ...). This is because only the difference of three in the pixel intensity affects the computation of the submapings when $\lambda > 0.1$ and it is difficult to get correct results.

4.2 The Histogram *h*(*l*)

The checking of $C_f^{(m,s)}$ in Section 4.1.2 does not depend on the histogram $h(l)$. The checking of BC and the third condition of BC in Section 4.1.4 may depend on $h(l)$. Actually, k is typically around one if we plot $\left(\lambda, C_f^{(m,s)}\right)$. In the implementation, we have used $k = 1$; i.e., we check $B_0\lambda^2$ and $B_1\lambda^2$. If the true value of k is smaller than one, $B_0\lambda^2$ and $B_1\lambda^2$ is not a constant and increases gradually by the factor $\lambda^{(1-k)/2}$. When $h(l)$ is constant, for example, the factor is $\lambda^{1/2}$. We can absorb such a difference by setting the threshold $B_{0_{thres}}$ appropriately.

Next, let us compute typical forms of $h(l)$. Let us model the source image by a circular object with its center at (x_0, y_0) and radius r given by

$$p_{i,j} = \begin{cases} \frac{255}{r}\, c\left(\sqrt{(i-x_0)^2+(j-y_0)^2}\right) & \left(\sqrt{(i-x_0)^2+(j-y_0)^2} \le r\right) \\ 0 & (otherwise) \end{cases}$$

and the destination image by

$$q_{i,j} = \begin{cases} \frac{255}{r}\, c\left(\sqrt{(i-x_1)^2+(j-y_1)^2}\right) & \left(\sqrt{(i-x_1)^2+(j-y_1)^2} \le r\right) \\ 0 & (otherwise) \end{cases}$$

with its center at (x_1, y_1) and radius r. Let $c(x)$ be in the form of $c(x) = x^k$. The histogram $h(l)$ is then in the form of

$$h(l) \propto rl^k \quad (k \neq 0)$$

if the centers (x_0, y_0) and (x_1, y_1) are sufficiently far. When $k = 1$, the images represent objects with clear boundaries embedded in the backgrounds. When $k_1 = -1$, the images represent objects with vague boundaries. Note that r is affected by the resolution of the images; i.e., $r \propto 2^m$. That is why the factor 2^m is introduced in Section 4.1.4.

4.3 Dynamic Determination of η

The other parameter η can be also determined automatically in the same manner. Initially, we set $\eta = 0$, and the final mapping $f^{(n)}$ and the energy $C_f^{(n)}$ of the finest resolution is again computed. After it is over, η is increased by a certain value $\delta\eta$ and we compute the final mapping $f^{(n)}$ and the energy $C_f^{(n)}$ of the finest resolution. We repeat this process until we obtain the optimal value.

η represents the stiffness of the mapping because it is a weight of

$$E_{0(i,j)}^{(m,s)} = \left\| f^{(m,s)}(i,j) - f^{(m,s-1)}(i,j) \right\|^2$$

when computing $D_f^{(m,s)}$. When η is zero, $D_f^{(n)}$ is determined irrespective of the previous submapping and the present submapping can be elastically deformed and becomes too distorted. When η is very large, $D_f^{(n)}$ is completely determined by the previous submapping; i.e., the submappings are very stiff, and the pixels are mapped to the same locations. The resulting mapping is therefore an identity mapping. When the value of η decreases, $C_f^{(n)}$ decreases, which means a better match. When it goes beyond the optimal value, however, it begins to increase (see Fig. 4) for the same reason described in Section 4.1.2. For the sake of speed, we detect this phenomenon in the reversed way; we increase the value of η starting from zero. The optimum value of η with the minimum $C_f^{(n)}$ is obtained in this manner. Since there are small fluctuations, whether the obtained value of $C_f^{(n)}$ is minimum cannot be judged immediately. We have to search for the true minimum around the obtained candidate value in detail again with a smaller interval.

5 SUPERSAMPLING

The range of $f^{(m,s)}$ can be expanded to $\mathbf{R} \times \mathbf{R}$ to increase the degree of freedom when deciding the correspondence between the pixels. (**R** stands for the set of real numbers.) In this case, the intensity values of the pixels of the destination image is interpolated to provide $f^{(m,s)}$ with the intensity values at noninteger points $V\left(q^{(m,s)}_{f^{(m,s)}(i,j)}\right)$; i.e., supersampling is performed. In the implementation, $f^{(m,s)}$ is allowed to take integer and half integer values, and $V\left(q^{(m,s)}_{(i,j)+(0.5,0.5)}\right)$ is given by $\left(V\left(q^{(m,s)}_{(i,j)}\right) + V\left(q^{(m,s)}_{(i,j)+(1,1)}\right)\right)\Big/2$.

6 NORMALIZATION OF THE PIXEL INTENSITY OF EACH IMAGE

When the source and destination images contain quite different objects, we cannot use the raw pixel intensity to compute the mapping. For example, let us consider a case

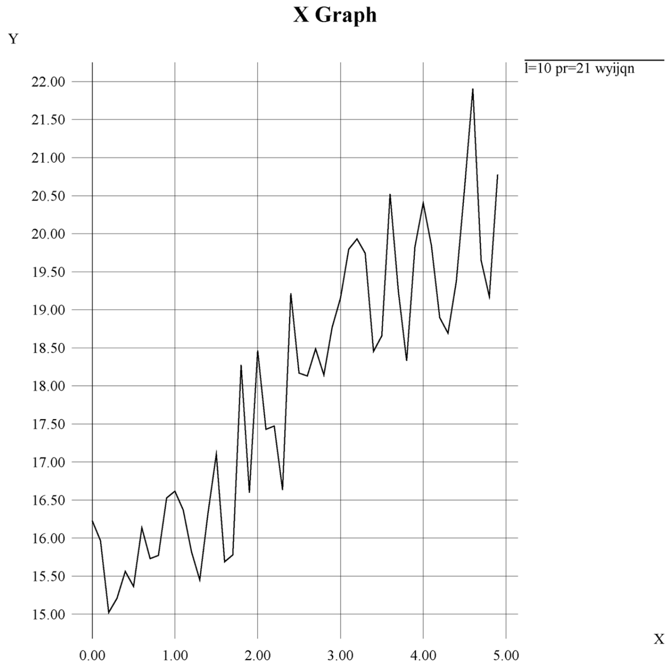


Fig. 4. An example graph showing the value of C_f (Y-axis) as η (X-axis) is varied.

where a human face and a face of a cat are matched as shown in Fig. 8. The face of the cat is furry and is a mixture of very bright pixels and very dark pixels. For this reason, the average pixel intensity of the subimages differ widely. In this case, to compute the submappings of the subimages of the facial images, it is necessary to normalize the subimages; i.e., the darkest pixel intensity should be set to zero, the brightest pixel intensity to 255, and the other pixel intensity values are linearly interpolated.

7 Implementation

In the implementation, we use a heuristic where the computation proceeds linearly as we scan the source image. First, the value of $f^{(m,s)}$ is determined at the top leftmost pixel $(i, j) = (0, 0)$. The value of each $f^{(m,s)}(i, j)$ is then determined while i is increased by one at each step. When i reaches the width of the image, j is increased by one and i is set to zero. In this way, $f^{(m,s)}(i, j)$ is determined while scanning the source image.

To avoid a bias, $f^{(m,s)}$ is determined starting from (0, 0) while increasing i and j when $s \bmod 4 = 0$, determined starting from the top rightmost location while decreasing i and increasing j when $s \bmod 4 = 1$, determined starting from the bottom rightmost location while decreasing i and j when $s \bmod 4 = 2$, and is determined starting from the bottom leftmost location while increasing i and decreasing j when $s \bmod 4 = 3$. At the finest level (the nth level) of resolution, we have used two values $s = 0$ and $s = 2$.

In the actual implementation, the values of $f^{(m,s)}(i, j)$ ($m = 0 .. n$) that satisfy BC are chosen from the candidates (k, l) by awarding a penalty to the candidates that violate BC. The energy $D_{(k,l)}$ of the candidate that violates the third condition is multiplied by φ and that of a candidate that violates the first or second condition is multiplied by ψ. In the implementation, $\varphi = 2$ and $\psi = 100{,}000$ are used.

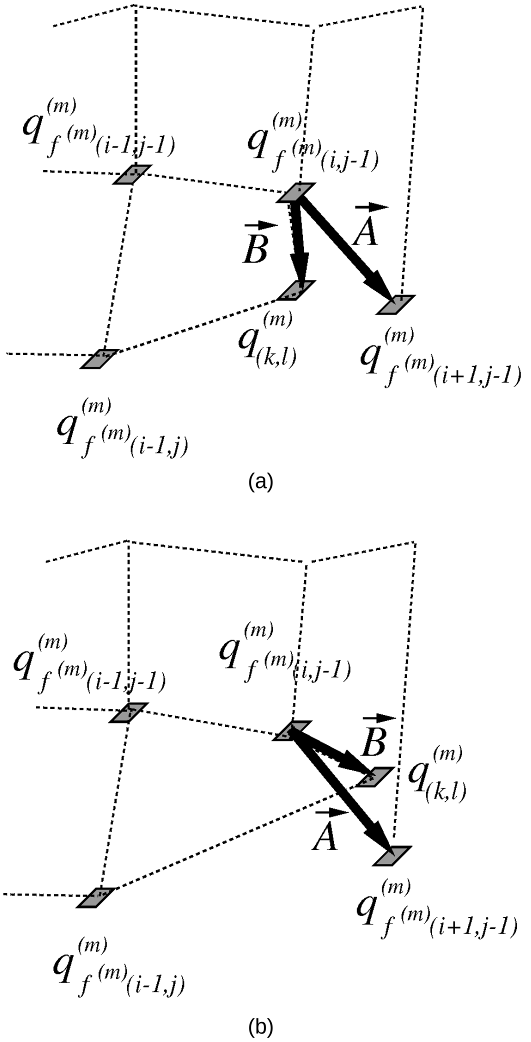

$q^{(m)}_{f^{(m)}(i-1,j-1)}$
$q^{(m)}_{f^{(m)}(i,j-1)}$
$\vec{B}$
$\vec{A}$
$q^{(m)}_{(k,l)}$
$q^{(m)}_{f^{(m)}(i+1,j-1)}$
$q^{(m)}_{f^{(m)}(i-1,j)}$
(a)
$q^{(m)}_{f^{(m)}(i-1,j-1)}$
$q^{(m)}_{f^{(m)}(i,j-1)}$
$\vec{B}$
$q^{(m)}_{(k,l)}$
$\vec{A}$
$q^{(m)}_{f^{(m)}(i+1,j-1)}$
$q^{(m)}_{f^{(m)}(i-1,j)}$
(b)

Fig. 5. Computing W to see if there is a pixel satisfying BC for $f^{(m)}(i, j + 1)$. (a) A candidate with no penalty. (b) A candidate with a penalty.

The actual procedure of checking BC includes the following test when determining $(k, l) = f^{(m,s)}(i, j)$. For each grid point (k, l) in the inherited quadrilateral of $f^{(m,s)}(i, j)$, we check whether the z-component of the outer product

$$W = \vec{A} \times \vec{B}$$

is equal to or greater than zero where

$$\vec{A} = \overrightarrow{q^{(m,s)}_{f^{(m,s)}(i,j-1)}\, q^{(m,s)}_{f^{(m,s)}(i+1,j-1)}}$$

and

$$\vec{B} = \overrightarrow{q^{(m,s)}_{f^{(m,s)}(i,j-1)}\, q^{(m,s)}_{(k,l)}}\,.$$

(Here, the vectors are regarded as 3D vectors and the z-axis is defined in the orthogonal right hand coordinate system.) When W is negative, the candidate is awarded a penalty by multiplying $D^{(m,s)}_{(k,l)}$ by ψ to avoid choosing it if possible.

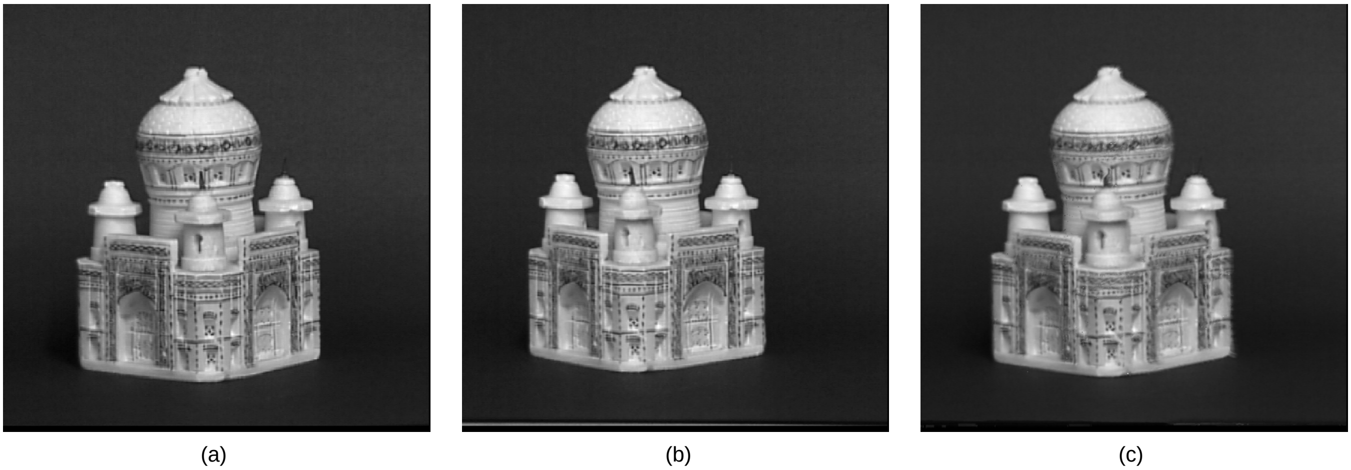


Fig. 6. Mapping of (a) the image of the left eye, (b) that of the right eye, and (c) the intermediate result of the interpolation.

The reason we check the above condition is as follows. When determining the mapping $f^{(m,s)}(i, j+1)$ for the adjacent pixel at $(i, j + 1)$, there is no pixel on the source image plane that satisfies BC if the z-component of W is negative. It is because $q^{(m,s)}_{(k,l)}$ exceeds the boundary of the adjacent quadrilateral (see Fig. 5).

7.1 The Order of Submappings

In the implementation, $\sigma(0) = 0$, $\sigma(1) = 1$, $\sigma(2) = 2$, $\sigma(3) = 3$, and $\sigma(4) = 0$ have been used at the even levels of resolution, and $\sigma(0) = 3$, $\sigma(1) = 2$, $\sigma(2) = 1$, $\sigma(3) = 0$, and $\sigma(4) = 3$ have been used at the odd levels of resolution.

8 Interpolations

After the mapping between the source and destination images is determined, the intensity values of the corresponding pixels are interpolated. In our implementation, we have used trilinear interpolation. Suppose a square $p_{(i,j)}p_{(i+1,j)}p_{(i+1,j+1)}p_{(i,j+1)}$ on the source image plane is mapped to a quadrilateral $q_{f(i,j)}q_{f(i+1,j)}q_{f(i+1,j+1)}q_{f(i+1,j+1)}$ on the destination image plane. For simplicity, the distance between the image planes is assumed to be one. The intermediate image pixels $r(x, y, t)$ $(0 \le x \le N - 1, 0 \le y \le M - 1)$ whose distance from the source image plane is $t(0 \le t \le 1)$ are obtained as follows. First, the location of the pixel $r(x, y, t)$ where $x, y, t \in \boldsymbol{R}$ is determined by the equation

$$\begin{aligned}(x, y) = {} & (1 - dx)(1 - dy)(1 - t)(i, j) + (1 - dx)(1 - dy)tf(i, j) \\ & + dx(1 - dy)(1 - t)(i + 1, j) + dx(1 - dy)tf(i + 1, j) \\ & + (1 - dx)dy(1 - t)\,(i, j + 1) + (1 - dx)dytf(i, j + 1) \\ & + dxdy(1 - t)(i + 1, j + 1) + dxdytf(i + 1, j + 1).\end{aligned}$$

The value of the pixel intensity at $r(x, y, t)$ is then determined by the equation

$$\begin{aligned}V(r(x, y, t)) = {} & \\ & (1 - dx)(1 - dy)(1 - t)V(p_{(i,j)}) + (1 - dx)(1 - dy)tV(q_{f(i,j)}) \\ & + dx\,(1 - dy)(1 - t)V(p_{(i+1,j)}) + dx\,(1 - dy)tV(q_{f(i+1,j)}) \\ & + (1 - dx)dy(1 - t)V(p_{(i,j+1)}) + (1 - dx)dytV(q_{f(i,j+1)}) \\ & + dxdy(1 - t)V(p_{(i+1,j+1)}) + dxdytV(q_{f(i+1,j+1)})\end{aligned}$$

where dx and dy vary from zero to one.

9 Application Examples

We can interpolate various images using our method. When two images from different viewpoints are interpolated, we can generate the images from intermediate viewpoints. It has strong applications in WWW because it enables us to generate arbitrary views from a limited number of images. When we interpolate images of two persons' faces, the interpolation gives us their morphing. When the images are cross-sections of 3D objects such as CT and MRI data, the interpolation enables us to reconstruct accurate 3D object shapes for volume rendering.

When the source and the destination images are color images, they are first converted to monochrome images, and the mappings are then computed. The original color images are then transformed by the resulting mappings.

Fig. 6 shows the case where the mapping is used for generating intermediate views. The image of the left eye and that of the right eye are interpolated. Each image size is 512 × 512. Fig. 6a shows the source image of the left eye, Fig. 6b shows that of the right eye, and Fig. 6c is the resulting intermediate image where the value of the parameter t described in Section 8 is 0.5 for simplicity. This example shows a rigid motion where the building is translated and rotated. The motion is best visible by looking at the central pillar that moves about 32 pixels.

Fig. 7 shows the case where the mapping is used for morphing of human faces. Two face images of different persons are interpolated. Fig. 7a shows the source image, Fig. 7b shows the destination image, Fig. 7c is the source

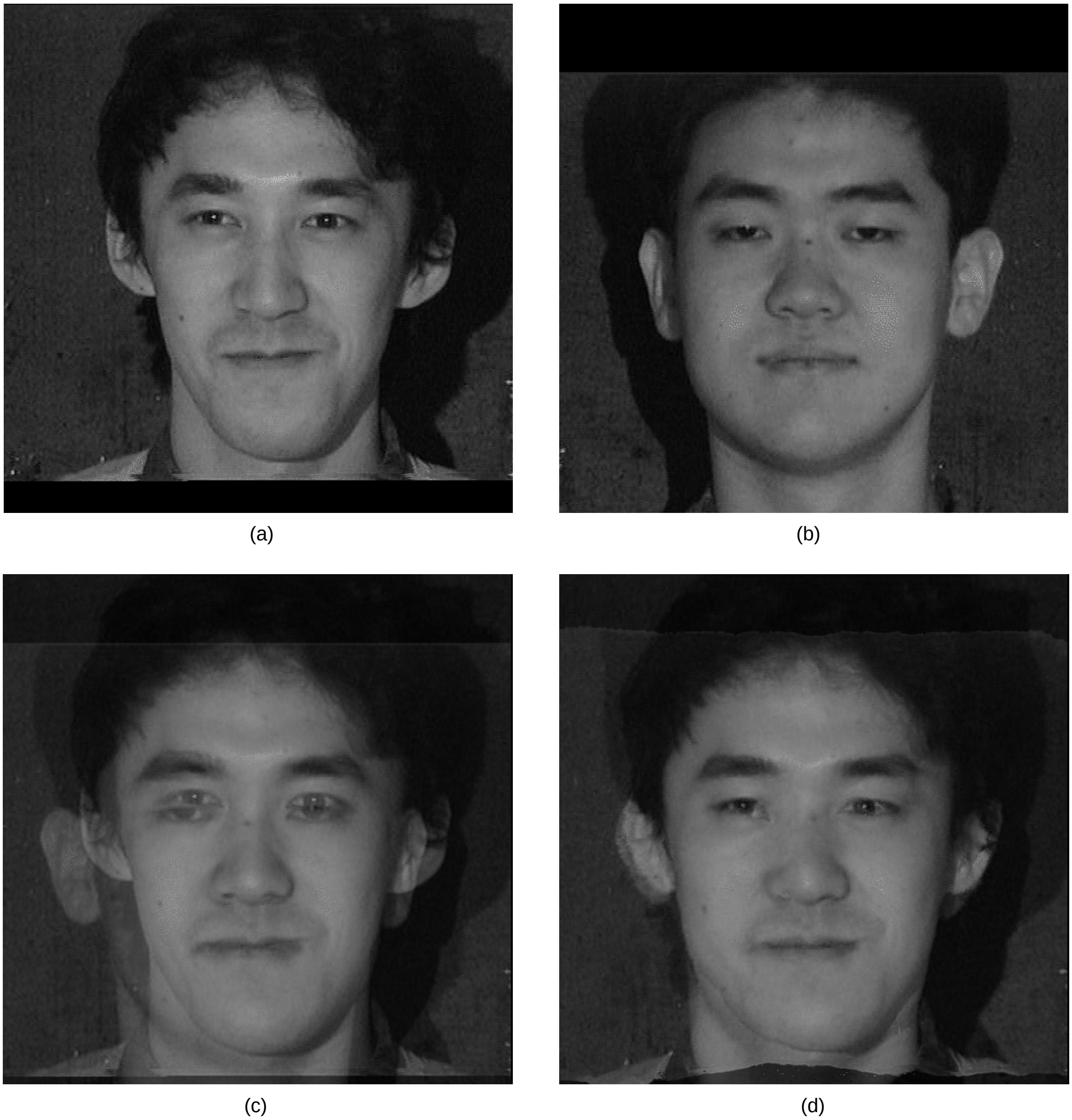


Fig. 7. Morphing of faces with (a) the source image, (b) the destination image, (c) superimposed image, and (d) the interpolated image.

image superimposed on the destination image, and Fig. 7d is the resulting intermediate image where $t = 0.5$.

Fig. 8 shows the case where the mapping is used for interpolating a human face and a face of a cat. Fig. 8a shows the source image, Fig. 8b shows the destination image, Fig. 8c is the resulting intermediate image where $t = 0.5$. The normalization of the pixel intensity described in Section 6 is used in this example only.

Fig. 9 shows the case where the method is applied to images with a number of objects. Each image size is 512 × 512. The motion is a translation parallel to the image plane. It can be best visible by looking at the air conditioner that moves about 53 pixels horizontally. Fig. 9a shows the source image, Fig. 9b shows the destination image, Fig. 9c is the resulting intermediate image where $t = 0.5$.

Fig. 10a shows the result where the mapping is used for interpolating images of a human brain whose cross-sections are obtained by MRI. Fig. 10a shows the source image, Fig. 10b the destination image (the upper cross section), Fig. 10c the interpolated image where the value of the parameter t is 0.5, and Fig. 10d shows the oblique view of the result of the volume rendering with four cross sections. The object is completely opaque and the interpolated pixels whose intensity is larger than $51 = 255 * 0.2$ are displayed. The reconstructed object is then cut vertically near the center to show the interior of the volume.

Fig. 11 shows the case where the source and destination images differ widely in shape. The cross-sectional images have been obtained by CT. Fig. 11a shows the source image, Fig. 11b the destination image, Fig. 11c the source image superimposed on the destination image, and Fig. 11d the interpolation result with $t = 0.5$.

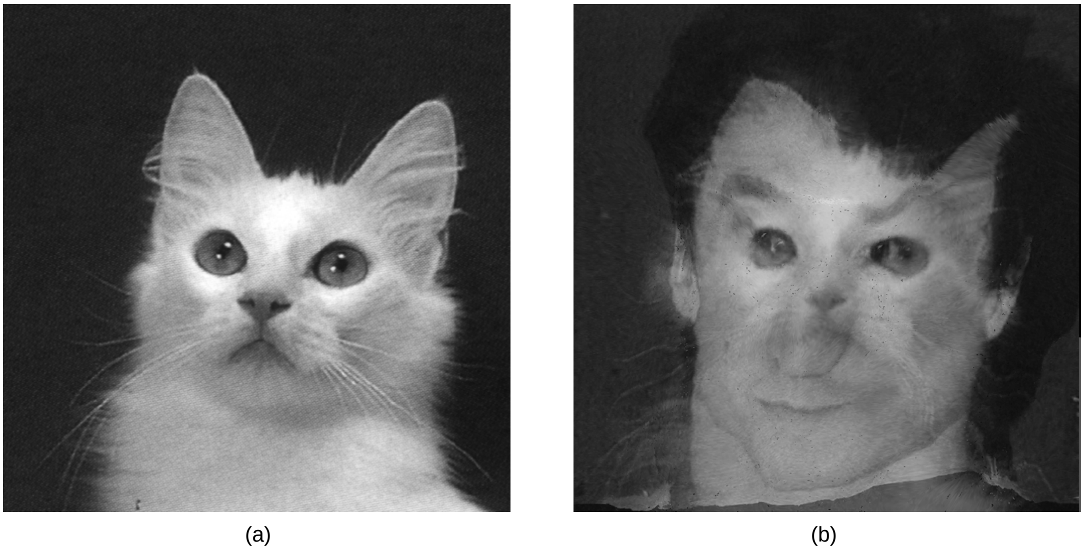


Fig. 8. Morphing of faces of a human face and (b) a face of a cat, with (a) the destination image, and (b) the interpolated image where the source image is shown in Fig. 7a.

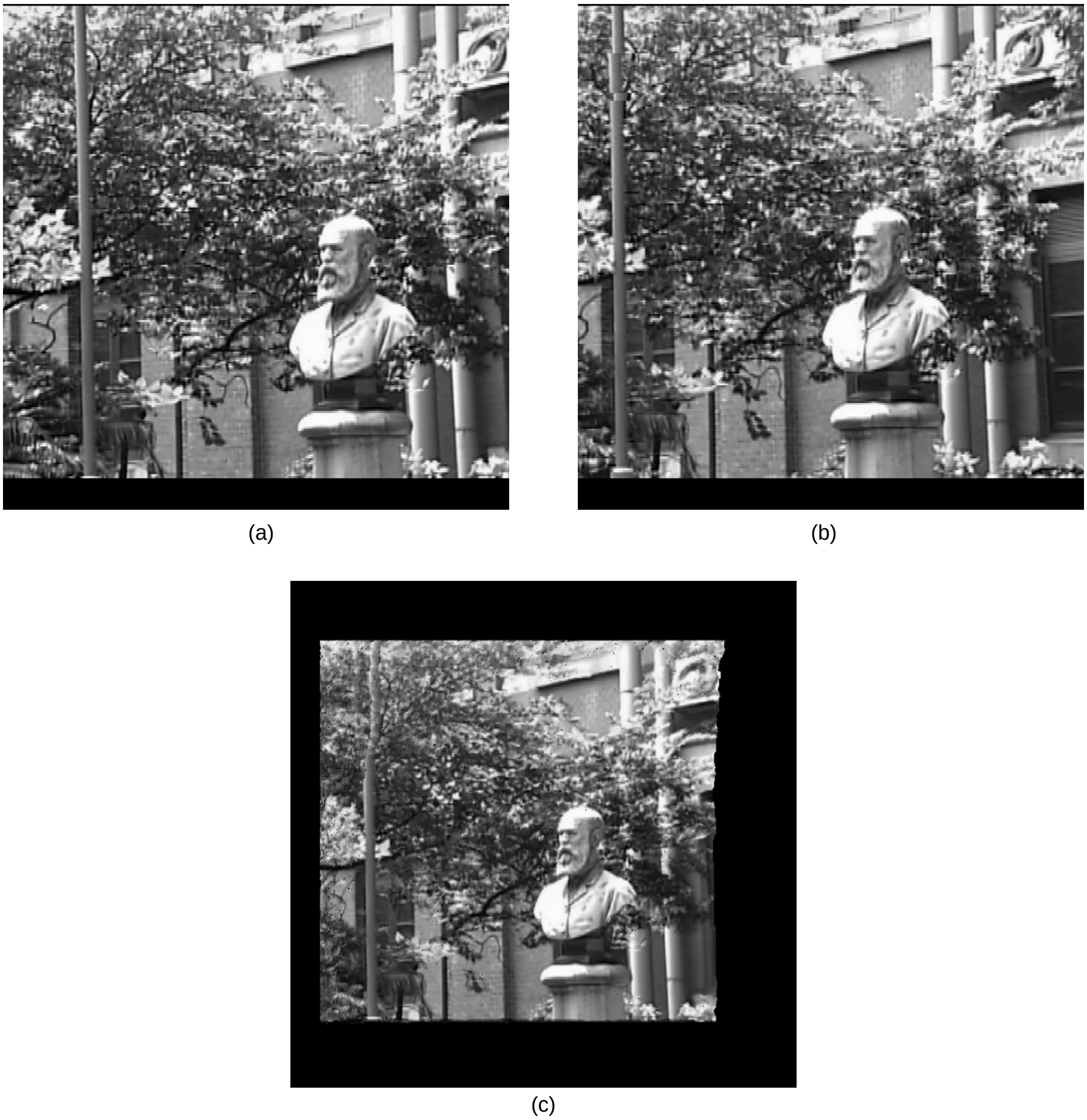


Fig. 9. Images with a number of objects. (a) The source image. (b) The destination image. (c) The interpolated image.

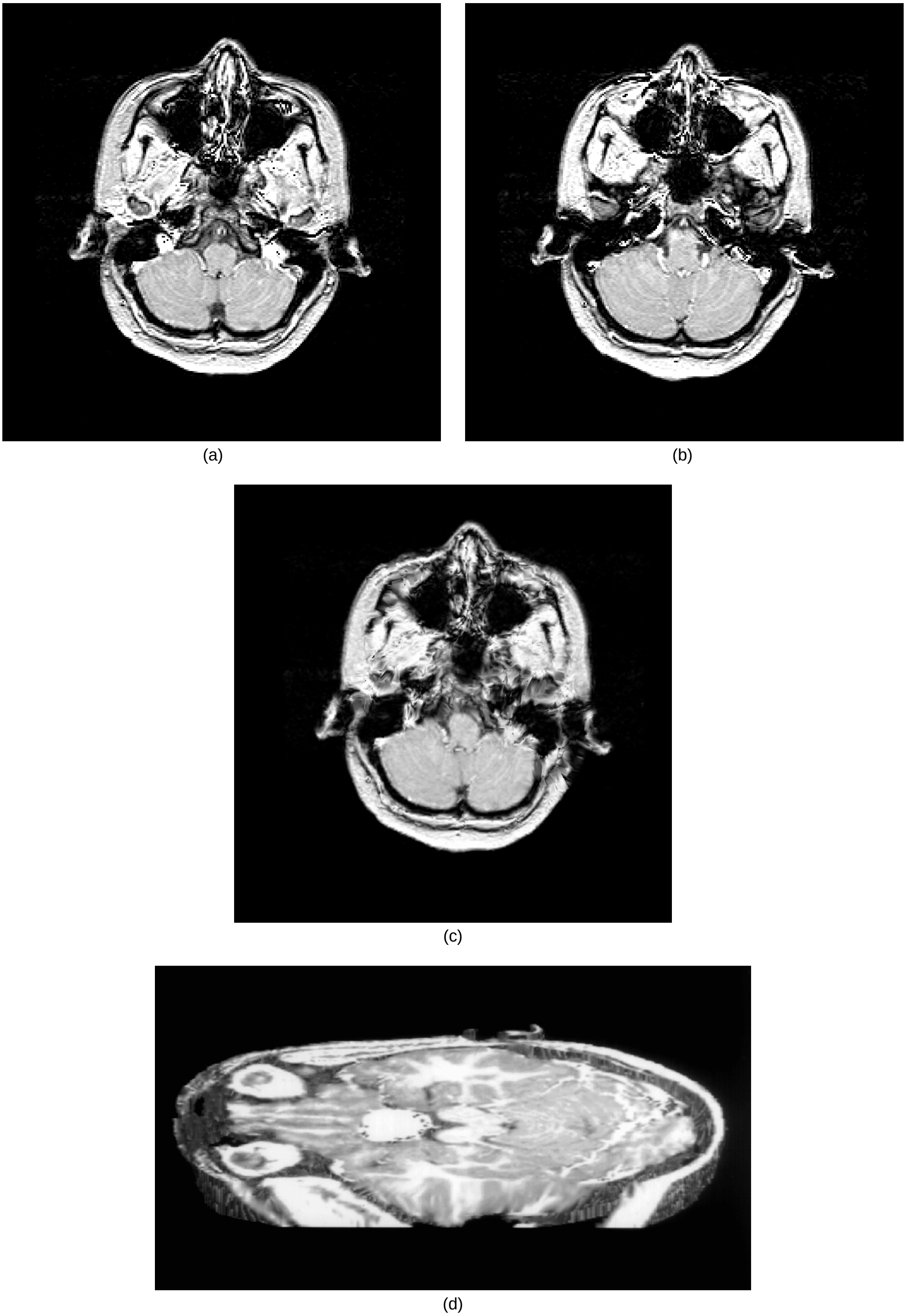
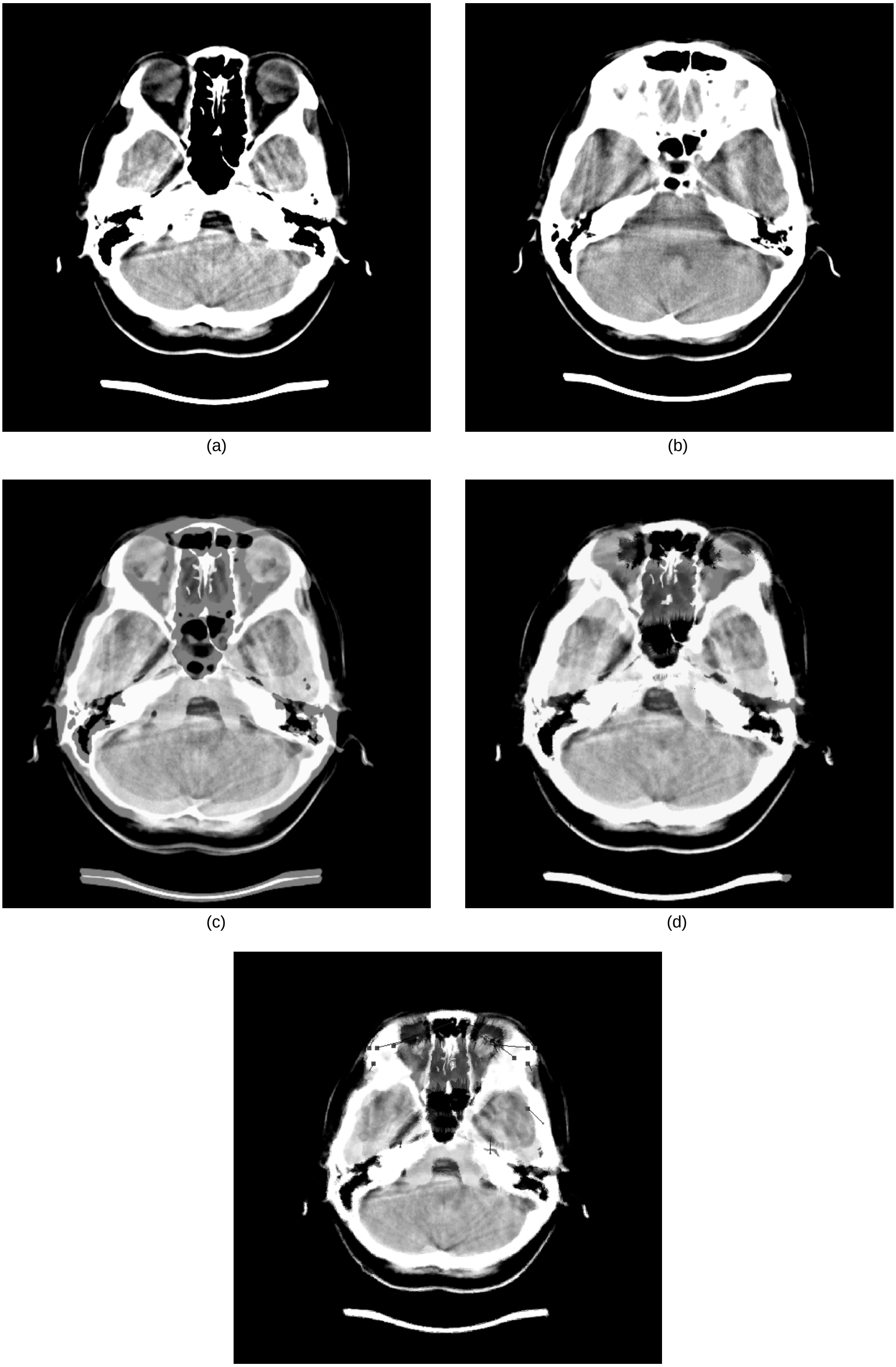


Fig. 10. Image interpolation of MR images with (a) the source image, (b) the destination image (the upper cross-section), (c) the interpolated image, and (d) the volume rendering with four cross-sections.



(a)

(b)

(c)

(d)

(e)

Fig. 11. The mapping where the source and destination images differ widely in shape with (a) the source image, (b) destination image, (c) their superimposed image, (d) the result without constraints, and (e) the result where we have specified the destinations of the 11 pixels in the source image in advance.

The size of each example image is 512 × 512 pixels except for that the size of each MRI image is 256 × 256 pixels. The intensity of a pixel varies from zero to 255. The SJ condition described in Section 3.1 is used in all the application examples except for Fig. 9. In all the application examples, we have used $B_{0_{thres}} = 0.003$ and $B_{1_{thres}} = 0.5$, and it has not been necessary to modify their values for any image. The pixel intensity of each subimages have been normalized in Fig. 8 only.

10 Discussions

10.1 Comparison With Optical Flow

Our mapping method has several things common with the optical flow (e.g., [18], [24], [14]). Given two images, optical flow detects the motion of objects (rigid bodies) in the images. It assumes that the intensity of each pixel of the objects does not change. It computes the motion vector (u, v) of each pixel by solving

$$\frac{\partial V}{\partial x}u + \frac{\partial V}{\partial y}v + \frac{\partial V}{\partial t} = 0$$

together with some additional conditions such as the smoothness of the vector field of (u, v). Here, V is the pixel intensity of the image.

Optical flow takes into account the difference in the intensity of pixels and smoothness like our method. It cannot, however, be used for image transformation because it does not take into account the global mappings of the images; i.e., it considers only the local motion of the objects. Optical flow cannot detect the global correspondence between images because it concerns only the local change of pixel intensity as can be seen in the above equation and systematic errors are conspicuous when the displacements are large [29]. Our method can detect global correspondence using the multiresolution hierarchy.

10.2 Constraining the Mapping

When there is correspondence between particular pixels of the source and destination images, and it should be considered when computing the mapping, we can specify the correspondence before starting the automatic computing of the entire mapping f.

The basic idea is to distort the source image roughly by the approximate mapping that maps the specified pixels of the source image to the specified pixels of the destination image, and then compute the accurate mapping f. First, we determine the approximate mapping denoted by $F^{(m)}$ in the following. It maps the specified pixels of the source image to the specified pixels of the destination image. Other pixels of the source image are mapped to appropriate locations; i.e., if they are close to one of the specified pixels, they are mapped to the locations near the position where the specified one is mapped. Let us denote the approximate mapping at the level m of the resolution hierarchy by $F^{(m)}$. We will discuss how to determine F and $F^{(m)}$ later.

Second, we change the energy $D^{(m,s)}_{(i,j)}$ of the candidate mappings f so that mappings similar to F have lower energy. To be precise, we have

$$D^{(m,s)}_{(i,j)} = E^{(m,s)}_{0(i,j)} + \eta E^{(m,s)}_{1(i,j)} + \kappa E^{(m,s)}_{2(i,j)}$$

where

$$E^{(m,s)}_{2(i,j)} = \begin{cases} 0, & \text{if } \left\|F^{(m)}(i,j) - f^{(m,s)}(i,j)\right\|^2 \le \left\lfloor \frac{\rho^2}{2^{2(n-m)}} \right\rfloor \\ \left\|F^{(m)}(i,j) - f^{(m,s)}(i,j)\right\|^2 & \text{otherwise} \end{cases}$$

and $\kappa; \rho \geq 0$ are real numbers.

Finally, the automatic computing process of mappings described before determines f completely.

Note that ${}^{(m,s)}E_{2_{(i,j)}}$ vanishes if $f^{(m,s)}(i, j)$ is sufficiently close to $F^{(m)}(i, j)$; i.e., if it is within the distance of $\left\lfloor \frac{\rho^2}{2^{2(n-m)}} \right\rfloor$. It is defined so because we want each value $f^{(m,s)}(i, j)$ to be determined automatically to fit in an appropriate place in the destination image so long as it is close to $F^{(m)}(i, j)$. Because of this, we do not have to specify the precise correspondence in detail; the source image is automatically mapped so that it matches the destination image.

The approximate mapping F is determined as follows. First, we specify the mapping for several pixels. When we have n_s pixels $p_{(i_0,j_0)}$, $p_{(i_1,j_1)}$, ..., and $p_{\left(i_{n_s-1},j_{n_s-1}\right)}$ of the source image to be specified, we determine the values

$$F^{(n)}(i_0, j_0) = (k_0, l_0),$$

$$F^{(n)}(i_1, j_0) = (k_1, l_1), \ldots,$$

and

$$F^{(n)}\left(i_{n_s-1}, j_{n_s-1}\right) = \left(k_{n_s-1}, l_{n_s-1}\right).$$

For the other pixels of the source image, the amount of displacement is the weighted average of the displacement of $p(i_h, j_h)$ ($h = 0, .., n_s - 1$); i.e, a pixel $p_{(i,j)}$ is mapped to the pixel of the destination image at

$$F^{(m)}(i,j) = \frac{(i,j) + \sum_{h=0}^{h=n_s-1}\left(k_h - i_h, l_h - j_h\right) weight_h(i,j)}{2^{n-m}}$$

where

$$weight_h(i,j) = \frac{1 \Big/ \left\|\left(i_h - i, j_h - j\right)\right\|^2}{total\ weight(i,j)}$$

and

$$total\ weight(i,j) = \sum_{h=0}^{h=n_s-1} 1 \Big/ \left\|\left(i_h - i, j_h - j\right)\right\|^2.$$

Instead of the method described above, we can use, for example, Beier and Neely's method [4] and specify line segments instead of specifying the pixels.

Fig. 11e shows the result where we have specified the destinations of the eleven pixels in the source image in advance. The locations of the specified pixels in the source image are represented by the larger square markers, and their destinations in the source image are represented by the smaller square markers being connected by line segments. We have used the parameter values $\kappa = 2.0$, $\rho = 32.0$ and for the Fig. 11e.

10.3 Semantic Correspondence and Robustness

As can be seen from Fig. 8, the correspondence between the noses of a man and a cat cannot be automatically computed because such correspondence cannot be inferred from the pixel intensity, but from the prior knowledge about the nose. The semantic correspondence can be specified by the aforementioned method.

10.3.1 Robustness

Suppose $p^{(m)}_{(i_0,j_0)}$ corresponds to $q^{(m)}_{(k_0,l_0)}$ semantically and the correspondence is correctly computed up to the mth level. When there is a pixel $q^{(m+1)}_{(k_1,l_1)}$ at the m + 1th level whose pixel intensity $V\left(q^{(m+1)}_{(k_1,l_1)}\right)$ is closer to $V\left(p^{(m+1)}_{(i_0,j_0)}\right)$, they may be connected instead. When η is close to zero, this can happen when

$$\lambda\left|V\left(q^{(m+1)}_{(k_1,l_1)}\right)-V\left(p^{(m+1)}_{(i_0,j_0)}\right)\right|^2 + \sum_{i'=i_0-1}^{i}\sum_{j'=j_0-1}^{j}\left\|\left((k_1,l_1)-(i_0,j_0)\right)-\left(f^{(m,s)}(i',j')-(i',j')\right)\right\|^2\Big/4$$
$$< \lambda\left|V\left(q^{(m+1)}_{(k_0,l_0)}\right)-V\left(p^{(m+1)}_{(i_0,j_0)}\right)\right|^2 + \sum_{i'=i_0-1}^{i}\sum_{j'=j_0-1}^{j}\left\|\left((k_0,l_0)-(i_0,j_0)\right)-\left(f^{(m,s)}(i',j')-(i',j')\right)\right\|^2\Big/4$$

holds by a rough estimation. From the inequality above, we can estimate the robustness as follows.

10.3.2 Semantic Correspondence That Requires Prior Knowledge of the Objects

By neglecting the terms $f^{(m,s)}(i', j') - (i', j')$, we can approximate the situation by

$$\lambda\left|V\left(q^{(m+1)}_{(k_1,l_1)}\right)-V\left(p^{(m+1)}_{(i_0,j_0)}\right)\right|^2 < \lambda\left|V\left(q^{(m+1)}_{(k_0,l_0)}\right)-V\left(p^{(m+1)}_{(i_0,j_0)}\right)\right|^2 + \left\|\left((k_1,l_1)-(k_0,l_0)\right)\right\|^2$$

i.e., when there is a pixel whose pixel intensity is closer to the source pixel in the vicinity of the destinations of the neighboring pixels than the semantically correct correspondence, our method chooses it. It means the case where the semantic correspondence is quite different from the correspondence inferred from the pixel intensity. Let us denote the value of the left hand side by X^2. Roughly speaking, the semantic correspondence can be correctly established if there is no pixel inside a ellipsoid in the ijV-coordinate system centered at $\left(k_0, l_0, V\left(p^{(m+1)}_{(i_0,j_0)}\right)\right)$ whose radius is $\sqrt{\lambda}X$ on the ij-plane and X along the V-axis. When there is another pixel inside the ball, it is connected instead of the pixel at (k_0, l_0). If we have to avoid the connection, we have to manually specify the correct one.

10.3.3 Outliers

Outliers such as noise can be analyzed in the same way. Suppose the pixel $p^{(m)}_{(i_0,j_0)}$ is an outlier that should be connected to $q^{(m)}_{(k_0,l_0)}$ for the sake of smoothness of the mapping. Let us assume that the correspondence is correctly computed up to the mth level. Again, it is wrongly connected to another pixel if that pixel is inside the ellipsoid described previously. If the noise is so strong as to appear in the coarse images (upper level of the hierarchy), the error becomes large.

In Fig. 6b, for example, there is noise (a horizontal white line) at the bottom of the destination image. The resulting mapping, however, is not affected by the noise because there is no corresponding pixel in the source image.

10.3.4 Nonrigid Transformation

Suppose the pixel intensity of the source and the destination images are similar but with nonrigid transformations. It is not difficult to compute this kind of transformation by our method as can be seen from Figs. 7 and 8.

10.4 Performance

For the MR images of Fig. 10 (256 × 256 pixels), our method takes for the computation from 20 to 30 seconds per mapping on an DEC Alphastation 333. For the images of Fig. 7 (512 × 512 pixels), our method takes from 10 to 15 minutes. As it takes $O(n^2)$ to compute a submapping of width n, the time complexity of our method is

$$O\left(1^2+4\left(2^2+4^2+8^2+\ldots+(N-1)^2\right)+2N^2\right)\approx$$
$$O\left(4\int_1^M 4^x\,dx-2N^2\right)$$
$$=O\left(\left((4\ln 4)-2\right)N^2\right)$$
$$\approx O\left(3.5N^2\right)\approx O\left(N^2\right)$$

where $N = 2^M$ is the width (height) of the images. That is, if we enlarge the images twofold, the time is quadrupled. The actual runtime increases, however, when the number of pixels violating BC becomes large, and it depends on the images.

11 Conclusions and Future Work

This paper has presented a new method for interpolating images automatically. New filters called the critical-point filters have been introduced. They enable the accurate matching of the critical points.

Development of a faster computation algorithm, particularly the search algorithm of the optimal parameters is left as a future research theme. Our method can be applied to the recognition of objects in images. The research on what kind of models should be prepared in advance to recognize the objects in the images is now progressing.

Handling occlusions on the boundaries of images is another future research theme. Because of the bijectivity condition, the pixels on the borders of the source image are mapped to the pixels on the borders of the destination image. This causes artifacts near the borders of the resulting images. It becomes noticeable when there are objects near the borders as can be seen in Fig. 9. The problem of occlusion is also conspicuous on the borders. When some objects on the border of the source image are not contained in the destination image, the system cannot find the corresponding pixels in the destination image, and hence the resulting interpolated image becomes slightly distorted on the border. When some objects in the destination image are not contained in the source image, they do not appear in the resulting interpolated image. In Fig. 9, for example, the left border is slightly distorted. This problem is serious when stereo photogrammetry is concerned [12]. When morphing is concerned, the occlusion problem is not noticeable because there are seldom occlusions in the given images. To remedy this, we may either cut off the borders or blur the resulting image on the borders when there are occlusions. The blurring is analogous to human visual systems because human eyes are of low resolution near the borders of sights.

Efficient direct computation of the mappings of color images is another future research theme. (In this paper, we converted them to monochrome images.) It can be basically achieved by computing submappings for the R (red), G (green), and B (blue) components. The computational cost, however, will be three times larger.

When there are bifurcations of regions in the images, it is useful to abandon BC at bifurcation points. Although our method can be easily expanded to handle bifurcations by omitting the one-to-one condition of the bijectivity, a more elegant theoretical framework is necessary for further complex applications.

ACKNOWLEDGMENTS

We would like to extend our gratitude to the members of the Shinagawa Laboratory of the University of Tokyo for fruitful discussions. Gratitude is extended to Dr. Etsuo Kunieda and Prof. Shozo Hashimito for providing us with medical images necessary for this research. Special thanks are extended to Nihon Sun Microsystems K. K. for offering Sparcstations, Hewlett-Packard Japan, Ltd. for HP9000, and Silicon Graphics Co., Ltd. for IRIS.

REFERENCES

[1] R. Bajcsy and S. Kovacic, “Multiresolution Elastic Matching,” *Computer Vision, Graphics, and Image Processing*, vol. 46, pp. 1-21, 1989.

[2] J.A. Bangham, P.D. Ling, and R. Harvey, “Scale-Space From Non-Linear Filters,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 18, no. 5, pp. 520-527, May 1996.

[3] G. Barequet and M. Sharir, “Partial Surface and Volume Matching in Three Dimensions,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 19, no. 9, pp. 929-948, Sept. 1997.

[4] T. Beier and S. Neely, “Feature-Based Image Metamorphosis,” *Computer Graphics (Proc. ACM SIGGRAPH’92)*, vol. 26, no. 2, pp. 35-42, July 1992.

[5] G. Borgefors, “Hierarchical Chamfer Matching: A Parametric Edge Matching Algorithm,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 10, no. 6, pp. 849-865, June 1988.

[6] G. Brookshire, M. Nadler, and C. Lee, “Automated Stereophotogrammetry,” *Computer Vision, Graphics, and Image Processing*, vol. 52, pp. 276-296, 1990.

[7] D.J. Burr, “A Dynamic Model for Image Registration,” *Computer Vision, Graphics, and Image Processing*, vol. 15, pp. 102-112, 1981.

[8] J. Canny, “A Computational Approach to Edge Detection,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 8, no. 6, pp. 679-698, 1986.

[9] C.K. Chui, *An Introduction to Wavelets*. Orlando, Fla.: Academic Press, 1992.

[10] I. Daubechies, *Ten Lectures on Wavelets*. Philadelphia, Penn.: Soc. Industrial and Applied Mathematics, 1992.

[11] O. Faugeras and L. Robert, “What Can Two Images Tell Us About a Third One,” *Int’l J. Computer Vision*, vol. 18, pp. 5-19, 1996.

[12] D. Geiger, B. Ladendorf, and A. Yuille, “Occlusions and Binocular Stereo,” *Int’l J. Computer Vision*, vol. 14, pp. 211-226, 1995.

[13] W.E.L. Grimson, “An Implementation of a Computational Theory of Visual Surface Interpolation,” *Computer Vision, Graphics, and Image Processing*, vol. 22, pp. 39-69, 1983.

[14] N.C. Gupta and L.N. Kanal, “3-D Motion Estimation From Motion Field,” *Artificial Intelligence*, vol. 78, pp. 45-86, 1995.

[15] M. Herman and T. Kanade, “Incremental Reconstruction of 3D Scenes From Multiple, Complex Images,” *Artificial Intelligence*, vol. 30, pp. 289-341, 1986.

[16] M. Herman, T. Kanade, and S. Kuroe, “Incremental Acquistion of a Three-Dimensional Scene Model From Images,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 6, no. 3, pp. 331-340, 1984.

[17] W. Hoff and N. Ahuja, “Surfaces From Stereo: Integrating Feature Matching, Disparity Estimation, and Contour Detection,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 11, no. 2, pp. 121-136, Feb. 1989.

[18] B.K.P. Horn, *Robot Vision*. Cambridge, Mass., The MIT Press, 1986.

[19] M. Levoy, “Display of Surfaces From Volume Data,” *IEEE Computer Graphics and Applications*, vol. 8, no. 3, pp. 29-37, 1988.

[20] W.E. Lorensen and H.E. Cline, “Marching Cubes: A High Resolution 3D Surface Construction Algorithm,” *Computer Graphics (Proc. ACM SIGGRAPH’87)*, vol. 21, no. 4, pp. 163-169, Aug. 1987.

[21] S.G. Mallat, “A Theory for Multiresolution Signal Decomposition: The Wavelet Representation,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 11, no. 7, pp. 674-693, July 1989.

[22] Y. Meyer, *Wavelets: Algorithms and Applications*. Philadelphia, Penn.: Soc. Industrial and Applied Mathematics, 1993.

[23] T. Nishita, T. Fujita, and E. Nakamae, “Metamorphosis Using Bézier Clipping,” T.S. Chua and T.L. Kunii, eds., *Mulitimedia Modeling (Proc. First Int’l Conf. Multimedia Modeling)*, pp. 162-175. Singapore: World Scientific, 1993.

[24] M. Otte and H.-H. Nagel, “Estimation of Optical Flow Based on Higher-Order Spationtemporal Derivatives in Interlaced and Noninterlaced Image Sequences,” *Artificial Intelligence*, vol. 78, pp. 5-43, 1995.

[25] S.M. Seitz and C.R. Dyer, “View Morphing,” *Proc. ACM SIGGRAPH’96*, pp. 21-30, 1996.

[26] D. Terzopoulos, “The Computation of Visible-Surface Representation,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 10, no. 4, pp. 417-438, 1988.

[27] S. Ullman and R. Basri, “Recognition by Linear Combinations of Models,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 13, no. 10, pp. 992-1,006, Oct. 1991.

[28] V. Venkateswar and R. Chellappa, “Occlusions and Binocular Stereo,” *Int’l J. Computer Vision*, vol. 15, pp. 245-269, 1995.

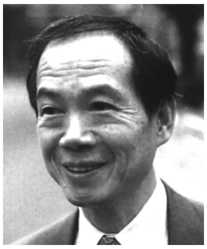
[29] J. Weber and J. Malik, “Robust Computation of Optical Flow in a Multi-Scale Differential Framework. *Int’l J. Computer Vision*, vol. 14, pp. 67-81, 1995.

[30] G. Wolberg, *Digital Image Warping*. Los Alamitos, Calif.: IEEE CS Press, 1990.

[31] Y. Yang and A.L. Yuille, “Multilevel Enhancement and Detection of Stereo Disparity Surfaces,” *Artificial Intelligence*, vol. 78, pp. 121-145, 1995.

[32] Z. Zhang, R. Deriche, O. Faugeras, and Q-T. Luong, “A Robust Technique for Matching Two Uncalibrated Images Through the Recovery of the Unknown Epipolar Geometry,” *Artificial Intelligence*, vol. 78, pp. 87-119, 1995.

Yoshihisa Shinagawa received his BSc (1987), MSc (1990), and DSc (1992) degrees in information science from the University of Tokyo. He is currently assistant professor of the Department of Information Science at the University of Tokyo. His research interests include computer graphics, vision, and its applications. He has published more than 50 refereed academic/technical papers in computer science. He is associate editor-in-chief of *The International Journal of Shape Modeling* (World Scientific). He is a member of the IEEE Computer Society, ACM, IPSJ, and IEICE.



Tosiyasu L. Kunii received his BSc in 1962, MSc in 1964, and DSc in 1967 all from the University of Tokyo. He had been Professor of Computer and Information Science at the University of Tokyo until March, 1993. He was the founding president and professor of the University of Aizu. He is currently professor emeritus of the University of Tokyo, professor of Hosei University, adviser of Fukushima Prefecture in Science and Technology, liaison counselor of Fukushima Prefecture Industrial Technology Foundation, and senior partner of Monolith. He has authored and edited around 50 books in computer science and also in general areas, and published more than 250 refereed academic/technical papers in computer science and applications. Dr. Kunii is founder of the Computer Graphics Society, editor-in-chief of *The Visual Computer: An International Journal of Computer Graphics* (Springer-Verlag) (1984-) and formerly *International Journal of Shape Modeling* (World Scientific) (1994-1995), associate editor-in-chief of *The Journal of Visualization and Computer Animation* (John Wiley & Sons) (1990-), and on the editorial boards of *Information Systems Journal* (1976-), *Information Sciences Journal* (1983-), and *IEEE Computer Graphics and Applications* (1982-).