J. Korean Soc. Math. Educ. Ser. B: Pure Appl. Math. ISSN 1226-0657
Volume 17, Number 4 (November 2010), Pages 321–332

A COMPARISON OF RADIAL BASIS FUNCTIONS IN APPLICATIONS TO IMAGE MORPHING

Boram Jin[a] and Yonghae Lee[b]

Abstract. In this paper, we experiment image warping and morphing. In image warping, we use radial basis functions : Thin Plate Spline, Multi-quadratic and Gaussian. Then we obtain the fact that Thin Plate Spline interpolation of the displacement with reverse mapping is the efficient means of image warping. Reflecting the result of image warping, we generate two examples of image morphing.

1. Introduction

Image morphing is an animated transformation of one image into another. It consists of two steps; one is warping process that changes and distorts images in various ways like pulling, pushing in, and so on, and the other is cross-dissolving process that mixes the colors of the images [1]. When cross-disssolving alone is applied without warping process, it does not match the locations of feature points (e.g. nose, mouth, pupils and so on) and generates the double exposure defect [1]. Figure 1.1 illustrates the differences of image morphing with and without warping process. Image (c) is an intermediate image generated by morphing the two input images (a) and (b) without warping, and dislocates many feature points. On the other hand, image (d) is generated by morphing with warping, and matches the locations of feature points. Warping process is indespensible in image morphing to maintain geometric alignment throughout cross-dissolving [8].

Several methods have been employed in warping process such as affine transformation, mesh relocation, and radial basis function interpolation. Each method has advantages and limitations, and could be applied to different situations. Affine transformation is very simple to use and implement, but can be applied only when

Received by the editors May 20, 2010. Revised November 1, 2010. Accepted November 22, 2010.
2000 *Mathematics Subject Classification.* 68U10.
Key words and phrases. image warping, morphing, radial basis function, thin plate spline.

© 2010 The Korean Society of Mathematical Educucation

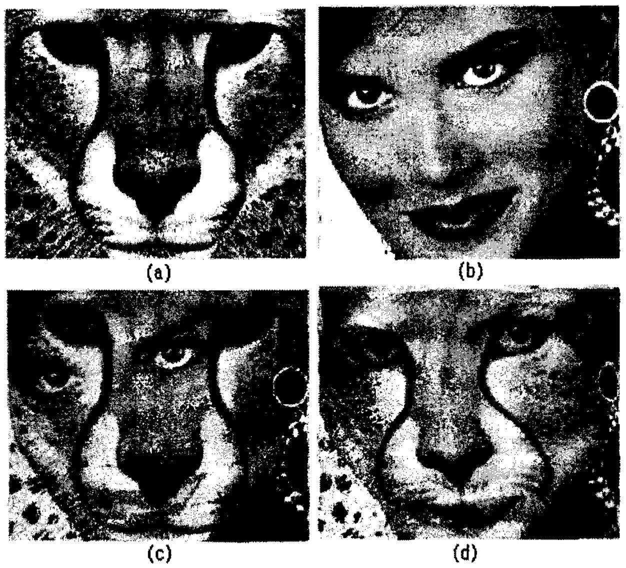

Figure 1.1. Images (a) and (b) are input pictures, image (c) results from a cross-disolving of the two input pictures, and image (d) from a morping.

warping effect is the same all over the images. Mesh relocation has simple specification of feature correspondence, but is cumbersome to use since a control mesh is always required for any structure of the image features[8]. Radial basis function interpolation can efficiently fit function sampled at generally scattered points, and is quite suitable in matching feature points. Hence we select radial basis function interpolation as a tool for warping process throughout this paper.

A radial basis function is defined as a function whose value depends only on the distance from a point. A radial basis function interpolation is simply the linear combination translates of basis functions such as multiquadratics(MQ), polyharmonic spline, thin-plate spline(TPS), gaussian and inverse multiquadrarics. Basis functions are classified into two groups; One is vanishes at infinity, and therefore its influence is local. The other does not vanish at infinity but rather increases unboundedly, and therefore its influence is global. Gaussian and inverse multiquadratics belong to the former case. MQ, polyharmonic spline and TPS belong to the latter case. We select Gaussian, MQ and TPS among them, since they are more accurate, stable, efficient and easy to implementation than the others [9, 5, 6]. While Gaussian and MQ have parameters that we need to determine experimentally, TPS leaves no free parameters.

In this paper, we experiment warping with three basis functions (TPS, Gaussian, MQ) and compare their output images. Then with the most convenient basis function among them, we go on to image morphing.

2. Radial Basis Functions

Here, we briefly review the RBF interpolations. Assume $X_1, \cdots, X_N$ are given scattered points in $\mathbb{R}^2$ and their corresponding $f_1, \cdots, f_N$ in $\mathbb{R}$ are given function values. Then we consider a function $R : \mathbb{R}^2 \to \mathbb{R}$ of the form

$$R(X) = \sum_{i=1}^{N} w_i \varphi(\|X - X_i\|) \tag{2.1}$$

that satisfies $R(X_i) = f_i$ for each i. Here $\|\cdot\|$ denotes the usual euclidean norm on $\mathbb{R}^2$, φ is a radial basis function and $w_1, w_2, \cdots w_N$ are corresponding coefficients of the basis function. From equation (2.1), we construct the following system of linear equations,

$$\Psi W = F$$

where $\Psi = [\varphi(\|X_i - X_j\|)]_{N \times N}$ is a interpolation matrix for the data points, $W = [w_1 w_2 \cdots w_N]^T$ is a coefficient vector and $F = [f_1 f_2 \cdots f_N]^T$ is a function value vector. If a basis function φ is one of the following two formulae,

$$\text{MQ} : \quad \varphi(r) = \sqrt{c^2 + r^2}$$
$$\text{Gaussian} : \quad \varphi(r) = e^{-\alpha r^2} \qquad (\alpha > 0)$$

then equation (2.1) is uniquely solvable for any distinct points [2].

$$\text{TPS} : \quad \varphi(r) = r^2 log(r)$$

For the TPS basis, however, equation (2.1) is not solvable in general[9], and requires an addition of linear polynomial $p(X)$.

$$R(X) = \sum_{i=1}^{N} w_i \varphi(\|X - X_i\|) + p(X) \tag{2.2}$$

where $p(X) = \lambda_0 + \lambda_1 x + \lambda_2 y$ for some constants $\lambda_0, \lambda_1$and λ_2. Equation (2.2) is uniquely solvable by the following augmented linear system.

$$\begin{bmatrix} \Psi & B \\ B^T & 0 \end{bmatrix} \begin{bmatrix} W \\ \Lambda \end{bmatrix} = \begin{bmatrix} F \\ 0 \end{bmatrix}$$

where $B = [1, x_i, y_i]$. The solution of TPS interpolation is known to minimize the total amount of bending energy [3], and would be the most commonly used one among the formentioned choices due to its robustness to oscillation. We solve equation (2.1) for basis function MQ and Gaussians; and equation (2.2) for TPS. The basis function $\varphi(r)$ of Gaussian rapidly approaches zero as r increases, so it is said to be local. On the other hand, the basis functions of MQ and TPS unboundedly increase as r increases, so they are said to be global.

3. IMAGE WARPING

In this section, we try the image warping based on the RBF interpolations (TPS, Gaussian and MQ). Unlike analog images, digital images are sampled on uniform lattice of finite resolution, and are represented as functions on the integer lattice $\mathbb{Z}^2$. Assume a source image $u_S(X) : \mathbb{Z}^2 \to \mathbb{R}$ is given, and some feature points $X_1, X_2, \cdots X_N$ are selected on it. The source points $X_1, X_2, \cdots X_N$ are moved to their respective destinations $A_1, A_2, \cdots, A_N$. An image warping finds a destination image $u_D(A) : \mathbb{Z}^2 \to \mathbb{R}$ preserving the movement of the feature points, i.e. $u_D(A_i) = u_S(X_i)$ for each i.

3.1. Forward or Reverse Mapping There are basically two implementations for the image warping. One is to find an forward mapping $f : \mathbb{R}^2 \to \mathbb{R}^2$ such that $f(X_i) = A_i$ for each i, and the other one is to find a reverse mapping $g : \mathbb{R}^2 \to \mathbb{R}^2$ such that $g(A_i) = X_i$ for each i. The mappings are used to generate the destination image as follows.

$$\text{Forward mapping} \quad u_D(f(X)) = u_S(X) \text{ for each } X \in \mathbb{Z}^2$$
$$\text{Reverse mapping} \quad u_D(A) = u_S(g(A)) \text{ for each } A \in \mathbb{Z}^2$$

The function values $f(X)$ and $g(A)$ may not fall on integer lattice point. In such cases, the point is rounded to its nearest integer point. The two mappings seem equivalent, however their effects are quite different. Figure 3.1 compares two mappings and illustrates that the forwad mapping maps all the four source points to one destination point, which results in the confusion about what to draw at the destination. Note that the confusion does not occur in reverse mapping. Further more, some lattice points of destination image may not be mapped by forward mapping, which causes some missing holes in its painting. Once again note that this does not happen in reverse mapping. Figure 3.2 depicts and compares typical outputs of forward and reverse mappings. Except for the mapping direction, the two images in the figure 3.2 were obtained in the same condition. The forward mapping generated many missing holes which can be seen as black circular scratches in the middle, and created confusions because it maps many source points to the same destination point.

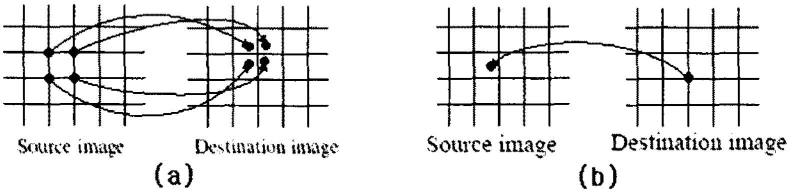

Figure 3.1. Images (a) and (b) illustrates forward mapping and reverse mapping, respectively. The forward mapping maps all the four source points to one destination point, which results in the confusion about what to draw at the destination.

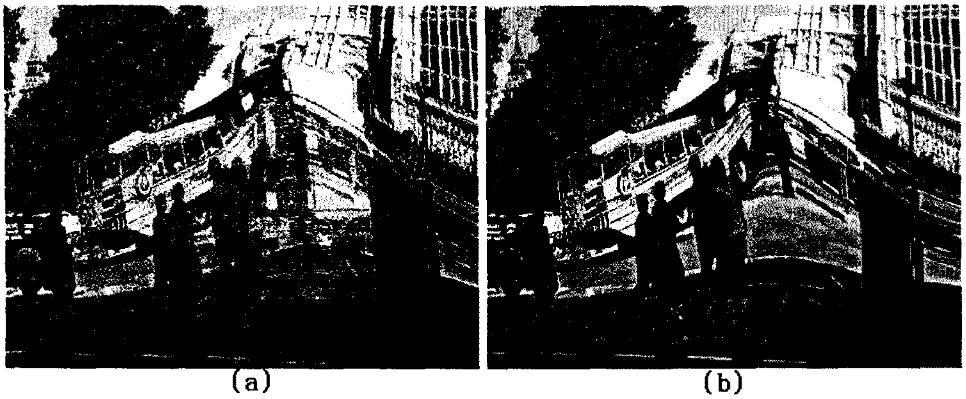

Figure 3.2. Images (a) and (b) were generated by a forward mapping and a reverse mapping, respectively. Except for the mapping direction, they were obtained in the same condition. The forward mapping generated many missing holes which can be seen as black circular scratches in the middle, and created confusions because it maps many source points to the same destination point; for conveniences the last source point in the raster scan was selected to resolve the confusion.

From these reasons, we use reverse mapping throughout this paper.

3.2. Interpolating Location or Displacement Now we consider finding the reverse mapping $g : \mathbb{R}^2 \to \mathbb{R}^2$ by RBF interpolations. There basically exists a reverse mapping which results from an RBF interpolation of the locations.

$$\text{Location interpolation} \quad R_l(A_i) = X_i \text{ for each } i = 1, 2, \cdots, N$$

We define a reverse mapping $g_l = R_l$. Using the map g_l, we experiment an example. For a given image (a) in figure 3.3, we choose $X_1, X_2, \cdots X_{25}$ in a uniform grid and set $A_i = X_i$ for each i. Note that R_l is vector-valued, hence the interpolation

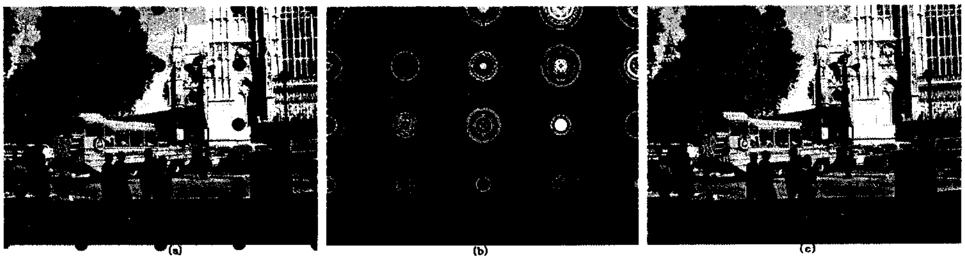

Figure 3.3. Image (a) is a source image. We choose 25 points in a uniform grid and set them as both of the source and destination points. 25 points are marked with ∘. Image (b) is an output image of map g_1 using Gaussian($\alpha = 0.001$). Note that Image (b) has the same color as the origin of image (a) far away from the points. Image (c) is the output image of map g_2 using Gaussian($\alpha = 0.001$). It is the same as source image (a).

algorithm in Section 2 is applied on each component. Since $X_i = A_i$, we generally expect that destination image would be the same as source image. But our expectaion is wrong in the case of using a local basis function. Image (b) in figure 3.3 is the destination image of map g_1 using Gaussian($\alpha = 0.001$). Let us consider a point A in image (b) far away from all A_i in image (a). Since $g_1(A) = R_1(A) \cong (0,0)$, $u_D(A) \cong u_S((0,0))$. So point A has the same color as the origin located at the bottom left. Hence the map g_1 is not a good choice for image warping.

So we consider another reverse mapping which results from an RBF interpolation of displacement instead of location[7].

$$\text{Displacement interpolation} \quad R_2(A_i) = X_i - A_i \text{ for each } i = 1, 2, \cdots, N$$

We define a reverse mapping $g_2 = R_2 + I$. Here, R_2 interpolates the displacement $X_i - A_i$ instead of the location X_i. We experiment the same example as before with the map g_2. Since $X_i - A_i = 0$, $R_2 = 0$ for all points in image. So the map $g_2 = R_2 + I = I$ and $u_D = u_S$. So, destination image agrees with source image as shown in image (c) in figure 3.3; and we have solved the problem of map g_1.

Now, we try a warping example illustrated in figure 3.4 with map g_2. To move up the yellow bus, we choose source points $X_1, X_2, \cdots X_9$ on the boundary of the bus; and we select their corresponding destination points $A_1, A_2, \cdots, A_9$ according to the upward movement of the bus. Once again note that R_2 is vector-valued, hence the interpolation algorithm in Section 2 is applied on each component.

Images (a), (b) and (c) in figure 3.5 are the destination images of map g_2 using TPS, Gaussian and MQ, respectively. In each of the images (a) and (c), the whole

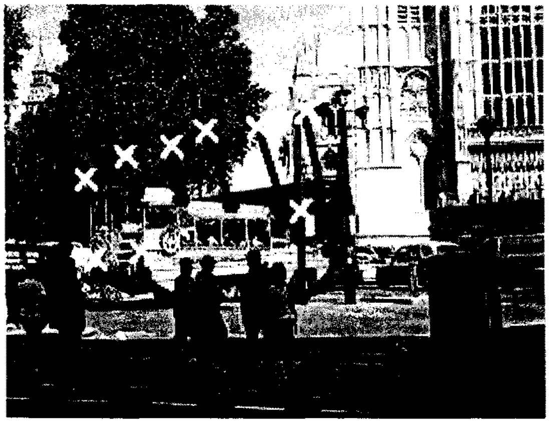

Figure 3.4. This is a source image for warping. 9 source points of the image are marked with ∘ and their corresponding destination points are marked with ×.

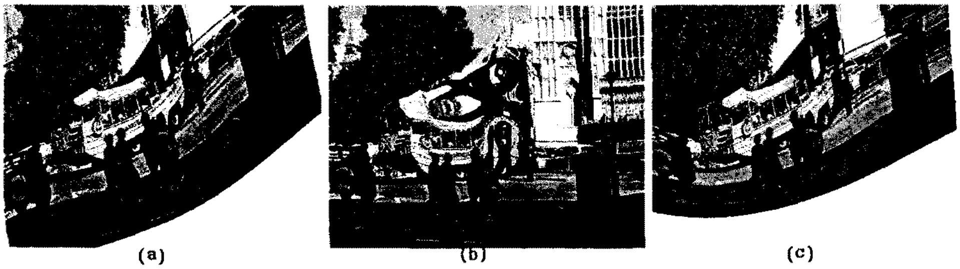

Figure 3.5. Images (a), (b) and (c) are the destination images of displacement interpolation using TPS, Gaussian($\alpha = 0.001$) and MQ, respectively. In each of the images (a) and (c), the whole image moves upward with the yellow bus. In image (b), the yellow bus does not maintain its shape and it mixes with the red bus.

image moves upward with the yellow bus. In image (b), the yellow bus does not maintain its shape and it mixes with the red bus.

So, we choose global basis functions for warping since we focus on the movement of the yellow bus.

3.3. Including Boundary Points To fix each edge of the images (a) and (c)in figure 3.5, we consider a simple method. As illustrated in figure 3.6, we choose some points on the boundary of the image and add them to both of the source and destination points.

Then we get two images (a) and (b) in figure 3.7. In each case, the boundary of the image is fixed and the yellow bus moves upward.

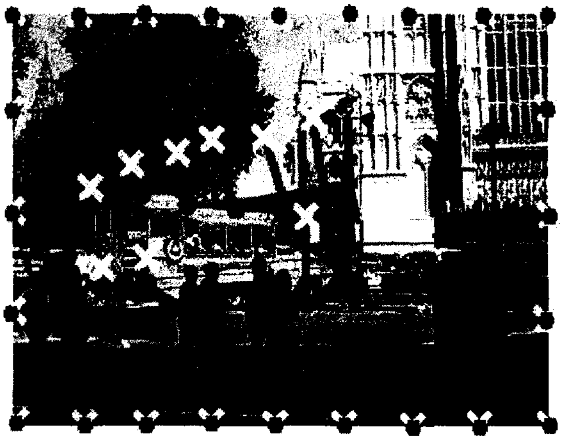

Figure 3.6. We choose 24 points on the boundary and add them to both of the source and destination points. Source points are marked with ∘ and their corresponding destination points are marked with ×.

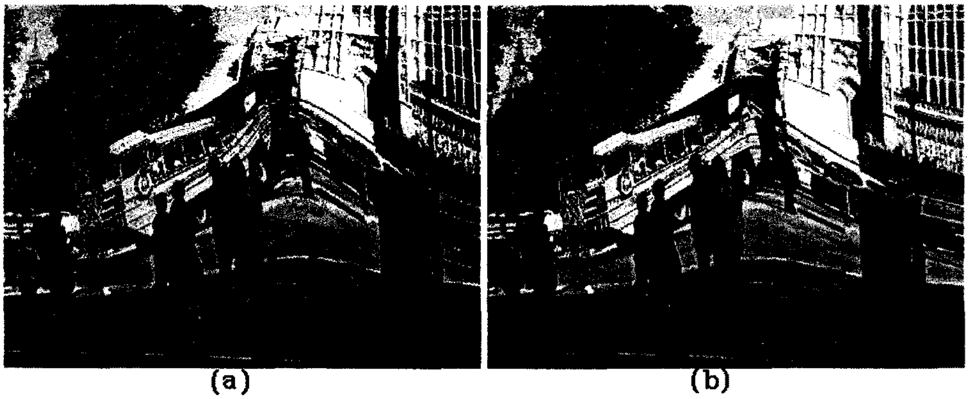

Figure 3.7. Including the 24 boundary points, we get images (a) and (b). They are the destination images of map g_2 using TPS and MQ, respectively.

3.4. Numerical Experiments Until now, we have experimented the example of image warping, as illustrated in figure 3.6. As a result, we get two images (a) and (b) in figure 3.7 close to our expectation; and they seem to be similar. However, MQ has a disadvantage to find a proper value of the parameter c, since a destination image depends on the value of the parameter c. For instance, images (a) and (b) in figure 3.8 are obtained in the same condition except for the value of the parameter c; $c = 0.1$ for image (a) and $c = 100$ for image (b).

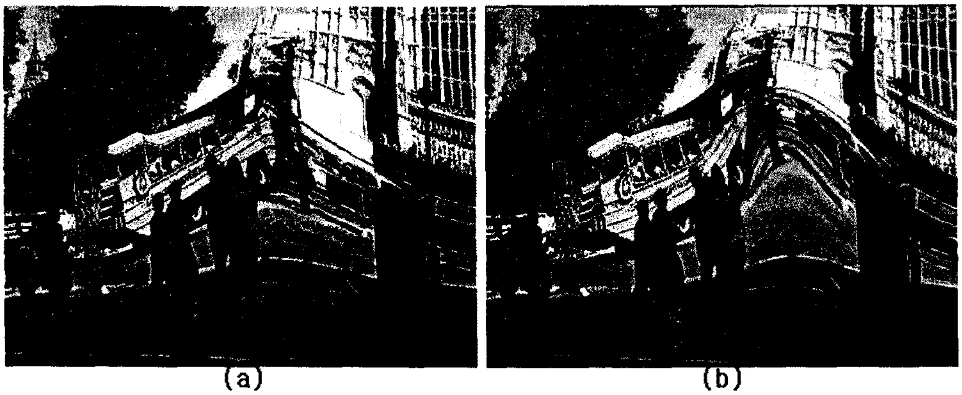

Figure 3.8. Images (a) and (b) in figure 3.8 are the destination images of map g_2 using MQ in the same condition except for the value of the parameter c. Image (a) and (b) are obtained with $c = 0.1$ and $c = 100$, respectively. Since a destination image depends on the value of the parameter c, image (b) differs from image (a).

From these reasons, we take TPS as the efficient means of image warping, and will use it throughout this paper.

4. Image Morphing

In this Section, we experiment image morphing. Morphing transforms one image into another by generating a series of intermediate synthetic images. Here, by the result of Section 3, we generate the intermediate images using the TPS-RBF interpolation of the displacement including boundary points.

For given source points $X_1, X_2, \cdots X_N$ and their corresponding destination points $A_1, A_2, \cdots, A_N$, we construct N lines from X_i to A_i for all i. Then, for each parameter $t \in (0, 1)$, lines intersect with the intermediate image at the intermediate points $I_1(t), I_2(t), \cdots, I_N(t)$ as illustrated in figure 4.1. Here, a point $I_i(t)$ is calculated as $I_i(t) = (1 - t) * X_i + t * A_i$. The source image u_S is warped into the image $u_S(t)$ by matching the source points $X_1, X_2, \cdots X_N$ to the intermediate points $I_1(t), I_2(t), \cdots, I_N(t)$. By the same way, the destination image u_D is warped into the image $u_D(t)$ by matching the destination points $A_1, A_2, \cdots, A_N$ to the intermediate points. The two warped images are cross-dissolved as [4]

$$u(t) = (1 - t) * u_S(t) + t * u_D(t) \tag{4.1}$$

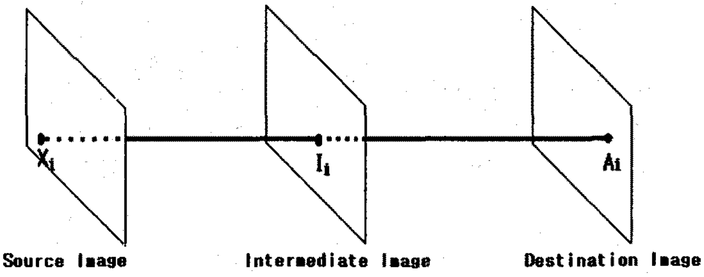

Figure 4.1. A line is drawn X_i in source image to A_i in destination image. For any $t \in (0,1)$, the intermediate point I_i is the intersection between the line and the intermediate image.

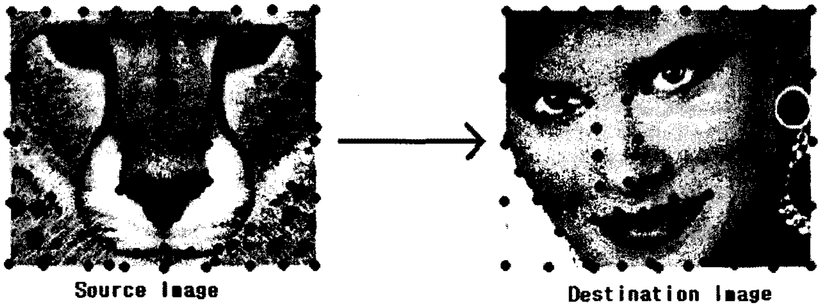

Figure 4.2. We choose the source image of leopard and the destination image of woman. Each 34 feature points and 24 boundary points in both of the source and destination points are marked with $\circ$.

Now, we experiment two examples using equation(4.1). One is that a leopard is morphed into a woman and the other is that an angora rabbit is morphed into a white lion. We select 34 feature points and 24 boundary points for the first example in figure 4.2 and we select 36 points feature points and 24 boundary points for the second example in figure 4.3. Then, we generate the intermediate images of two examples as shown in figure 4.4 and figure 4.5. Each source image smoothly transforms into the destination image with the series of intermediate synthetic images.

5. CONCLUSION

In this paper, we have experimented image warping and morphing. As described in Section 3, we have attempted to determine which method is most close to our expectation among RBF interpolations. As a result, we obtained the fact that the

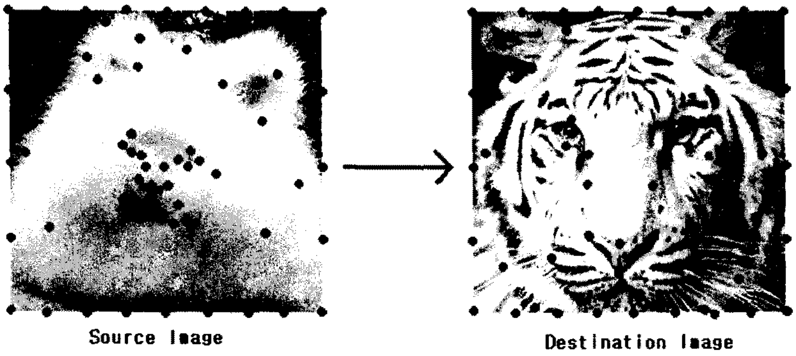

Figure 4.3. We choose the source image of angora rabbit and a destination image of white lion. Each 36 feature points and 24 boundary points in both of the source and destination points are marked with o.

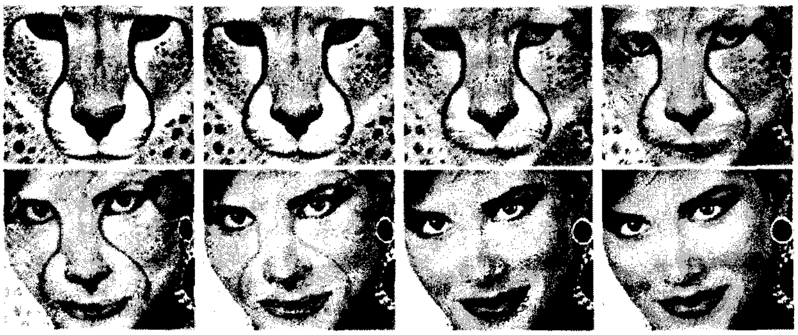

Figure 4.4. Intermediate images of image morphing from the leopard in left top to woman in right bottom.

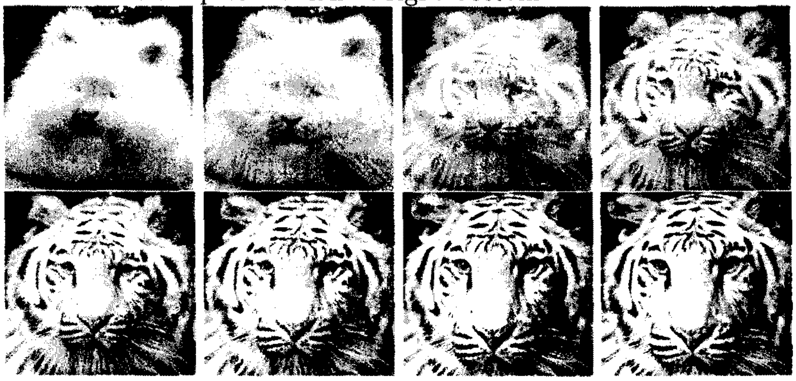

Figure 4.5. Intermediate images of image morphing from angora rabbit in the left top to lion in the right bottom.

TPS-RBF interpolation of the displacement with reverse mapping is the efficient means of image warping.

Reflecting the result of image warping, we obtained Equation(4.1) which contains two steps of image morphing, image warping and cross-dissolving. Then, using this equation, we generated two examples of image morphing as shown in figure 4.4 and figure 4.5.

REFERENCES

1. R. Crane: *A Simplified approach to Image Processing.* Prentice-Hall,1996.
2. N. Dyn: *Interpolation and approximation by radial and related functions, approximation theory.* Academic Press, 1989.
3. J. Duchon: *Spline minimizing rotation-invariant semi-norms in sobolev spaces, in constructive theory of functions of several variables.* Springer-Verlag, 1977.
4. H. Johan, Y. Koiso & T. Nishita: Morphing using curves and shape interpolation techniques. *IEEE Proceedings of the 8th Pacific Conference on Computer Graphics and Applications* (2000), 348-454.
5. R. Franke: Scattered data interpolation; test of some methods. *Math. of Comput.* **38** (1982), 181-199.
6. T. Li: Shear-warp rendering algorithm based on radial basis functions interpolation. *Comput. Model. and Simul.* **2** (2010), 425-429.
7. S. Lee & K. Chwa: Image morphing using deformable surface. *Proc. of the 7th Intl. conf. on comput. Anim., Switzerland* **200** (1994), 31-39.
8. S. Lee, G. Wolberg, K. Chaw & S. Shin: Image metamorphosis with scattered feature constraints. *IEEE Trans. on Visual. and Comput.* **2** (1996), 337-354.
9. N. Arad, N. Dyn & D. Reisfeld: Image warping by radial basis functions : application to facial expressions. *Graph. Models and Image Proc.* **56** (1994), 161-172. 1994

[a]DEPARTMENT OF MATHEMATICS, KYUNGHEE UNIVERSITY, SEOUL, KOREA 130-701
Email address: boramjin88@gmail.com

[b]DEPARTMENT OF MATHEMATICS, KYUNGHEE UNIVERSITY, SEOUL, KOREA 130-701
Email address: yonghaelee@gmail.com