



A Comparative Study of Transformation Functions for Nonrigid Image Registration

Lyubomir Zagorchev & Ardeshir Goshtasby

Abstract—Transformation functions play a major role in nonrigid image registration. In this paper, the characteristics of thin-plate spline (TPS), multiquadric (MQ), piecewise linear (PL), and weighted mean (WM) transformations are explored and their performances in nonrigid image registration are compared. TPS and MQ are found to be most suitable when the set of control-point correspondences is not large (fewer than a thousand) and variation in spacing between the control points is not large. When spacing between the control points varies greatly, PL is found to produce a more accurate registration than TPS and MQ. When a very large set of control points is given and the control points contain positional inaccuracies, WM is preferred over TPS, MQ, and PL because it uses an averaging process that smoothes the noise and does not require the solution of a very large system of equations. Use of transformation functions in the detection of incorrect correspondences is also discussed.

Index Terms—Image registration, transformation function, thin-plate spline, multiquadric, radial basis functions, piecewise linear, weighted-mean

1 Introduction

Image registration is a computational method for determining the point-by-point correspondence between two images of a scene. Many applications such as image fusion and change detection require point-by-point correspondence between images. Although nonrigid image registration has been achieved by optimizing an appropriate cost function without the use of control points [1, 36], most nonrigid registration methods first find a number of corresponding control points in the images

and then use information about the correspondences to find a transformation function that will determine the correspondence between all points in the images.

This paper focuses on the transformation function aspect of nonrigid image registration. Selection of control points in images and determination of the correspondence between them have been reported extensively elsewhere. Corner points [2, 13, 15, 27, 31, 43, 52, 53], region centroids [24], and line intersections [51] have been used as the control points, and correspondence between the control points have been found via chamfer matching [5, 6, 9, 32], graph matching [12], random sample consensus [14], probabilistic relaxation labeling [22, 42], matching of minimum-spanning-tree edges [59], matching of convex-hull edges [23], Hausdorff distance [30, 45], hierarchical attribute matching [49], clustering [51], template matching [21, 16, 54], Hough transform [58], and geometric hashing [56].

The performance of an image registration algorithm depends on 1) the performance of its control-point correspondence step and 2) the performance of the transformation function that uses information about the correspondences to warp one image to the geometry of the other. In this paper, the focus will be on the second step, that is, the determination of a transformation function from a given set of control-point correspondences. It is, therefore, assumed that a set of corresponding control points in the images is given. It is also assumed that the correspondences are correct although they may contain small positional inaccuracies. Later in the paper it will be shown how to detect the incorrect correspondences (outliers) from information in an obtained transformation function.

The problem to be addressed is as follows: Given the coordinates of N corresponding control points in two images of a scene,

$$\{(x_i, y_i), (X_i, Y_i) : i = 1, \dots, N\}, \quad (1)$$

determine a transformation function $\mathbf{f}(x, y)$ with components $f_x(x, y)$ and $f_y(x, y)$ that satisfy

$$X_i = f_x(x_i, y_i), \quad (2)$$

$$Y_i = f_y(x_i, y_i), \quad i = 1, \dots, N, \quad (3)$$

or

$$X_i \approx f_x(x_i, y_i), \quad (4)$$

$$Y_i \approx f_y(x_i, y_i), \quad i = 1, \dots, N. \quad (5)$$

Once $\mathbf{f}(x, y)$ is determined, given the coordinates of a point (x, y) in one image, the coordinates of the corresponding point (X, Y) in the other image can be determined. We will refer to the image with coordinates (x, y) as the *reference* and the image with coordinates (X, Y) as the *target*. The reference image is kept unchanged and the target image is resampled to take the geometry of the reference image.

If the coordinates of corresponding control points in the images are arranged into two sets of 3-D points as follows:

$$\{(x_i, y_i, X_i) : i = 1, \dots, N\}, \quad (6)$$

$$\{(x_i, y_i, Y_i) : i = 1, \dots, N\}, \quad (7)$$

then the components of the transformation as defined by equations (2) and (3) can be considered two explicit surfaces interpolating the two sets of 3-D points, and the components of the transformation defined by (4) and (5) can be considered explicit surfaces approximating the two sets of 3-D points. The components of the transformation for registering two 2-D images are, therefore, single-valued surfaces in 3-D. Since the two components of a transformation are similar, they can be determined in the same manner. Therefore, in the following, the problem of finding a single-valued surface $f(x, y)$ that interpolates or approximates

$$\{(x_i, y_i, f_i) : i = 1, \dots, N\} \quad (8)$$

will be addressed. It is required to determine the parameters of a transformation function that satisfies

$$f_i = f(x_i, y_i), \quad i = 1, \dots, N, \quad (9)$$

or

$$f_i \approx f(x_i, y_i), \quad i = 1, \dots, N. \quad (10)$$

Given a set of corresponding control points in reference and target images, many transformation functions can be found to accurately map the control points in the target image to the corresponding control points in the reference image. A proper transformation will accurately map the remaining points in the target image to the corresponding points in the reference image also. Some transformation functions, although mapping corresponding control points in the images to each other accurately, warp the target image too much, causing significant misregistration away from the control points. Also, since a transformation function is computed from the control point correspondences, inaccuracies in the correspondences will transfer to inaccuracies in the transformation function. It is desired that a transformation function will smooth noise and small inaccuracies in the correspondences. Therefore, when noise and inaccuracies in the correspondences exist, transformation functions that are based on approximation are preferred over transformation functions that are based on interpolation.

Various transformation functions have been used in image registration. In this paper, the properties of four popular transformation functions are examined. To determine the behaviors of the transformation functions and to find their accuracies in image registration, a number of test images as shown in Fig. 1 are used. Fig. 1a is an image of size 512×512 pixels containing a 32×32 uniform grid, and Fig. 1b shows the same grid after being translated by (5,8) pixels. Fig. 1c shows counterclockwise rotation of the grid by 0.1 radians about the image center, Fig. 1d shows scaling of the grid by 1.1 with respect to the image center, and Fig. 1e shows the grid after the following linear transformation

$$X = 0.7x - y + 3, \quad (11)$$

$$Y = 0.9x + 0.8y + 5. \quad (12)$$

Fig. 1f shows the grid after transformation by the following inverse distance function

$$X = x + 50(x - x_c)/r, \quad (13)$$

$$Y = y + 50(y - y_c)/r, \quad (14)$$

where $x_c = n_c/2$, $y_c = n_r/2$, n_r and n_c are the number of image rows and columns, respectively, and

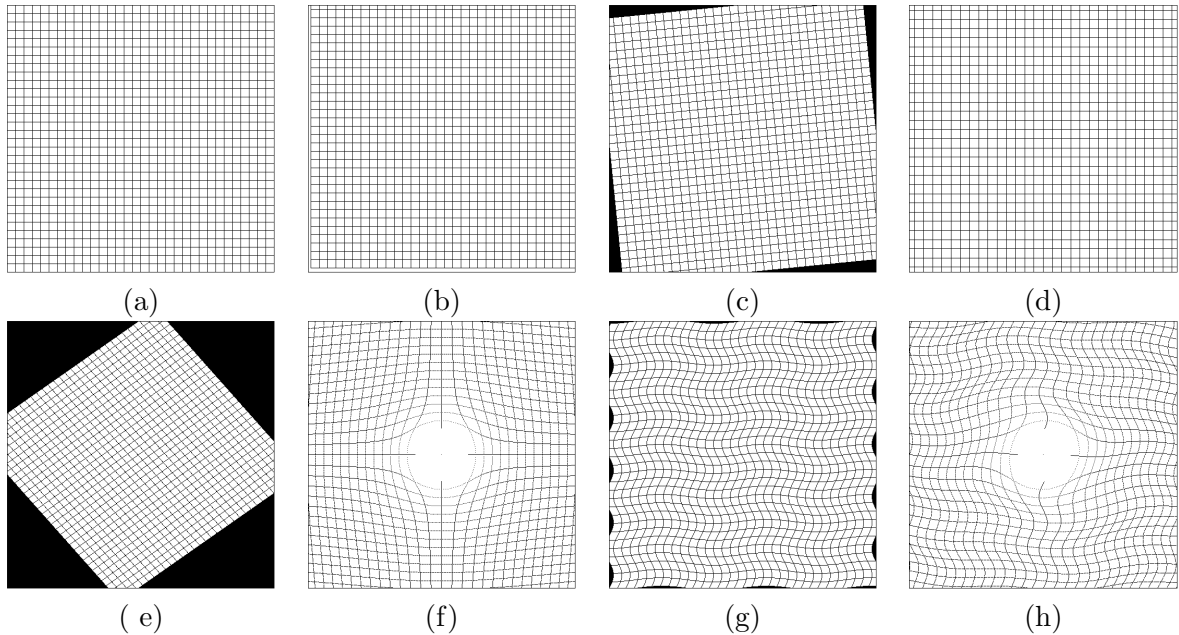


Figure 1: (a) A 32×32 uniform grid. The grid after (b) translation, (c) rotation, and (d) scaling. The grid after (e) linear, (f) inverse distance, (g) sinusoidal, and (h) a combination of sinusoidal and inverse distance transformations.

$r = \sqrt{(x - x_c)^2 + (y - y_c)^2}$. Fig. 1g shows transformation of the grid by the following sinusoidal function

$$X = x - 8 \sin(y/16), \quad (15)$$

$$Y = y + 4 \cos(x/32), \quad (16)$$

and Fig. 1h shows transformation of the grid by a combination of the inverse distance and sinusoidal:

$$X = x - 8 \sin(y/16) + 50(x - x_c)/r, \quad (17)$$

$$Y = y + 4 \cos(x/32) + 50(y - y_c)/r. \quad (18)$$

Translation and rotation represent rigid transformation and scaling represents a similarity transformation. The rigid, similarity, and linear transformations are used to determine the performances of a transformation function when the images do not have nonlinear geometric differences. An ideal transformation will not nonlinearly warp an image if the control point correspondences do not contain nonlinear geometric differences. The inverse distance function is used to find the effectiveness of a transformation in registration of images with global nonrigid differences. The sinusoidal function is used to find the effectiveness of a transformation in registration of images with local

geometric differences, and a combination of the inverse distance and sinusoidal functions is used to determine the effectiveness of a transformation in registration of images with both local and global geometric differences.

Different density and organization of point correspondences are used to estimate the geometric difference between Fig. 1a and Figs. 1b–h. Figure 2a shows all grid points in Fig. 1a, Fig. 2b shows a uniformly spaced subset of the grid points, Fig. 2c shows a random subset of the grid points, and Fig. 2d shows a subset of the grid points with variation in its density. Although uniformly spaced control points are rarely obtained in images, they are used here to determine the influence of the spacing of the control points on the registration accuracy. Seven geometric transformations are applied to the reference image to obtain the target images, and four sets of control points are used to estimate the transformation in each case. Therefore, overall, twenty eight cases are considered.

The control points in the reference image are the grid points shown in Figs. 2a–d. The control points in the target image are the corresponding grid points obtained by transforming them with the seven transformations and rounding the obtained coordinates. The rounding process displaces the control points in the target image from their true positions by up to half a pixel. From the coordinates of corresponding control points, the transformations are estimated and compared with the true transformations. In these experiments, although incorrect correspondences do not exist, corresponding control points contain rounding error (digital noise).

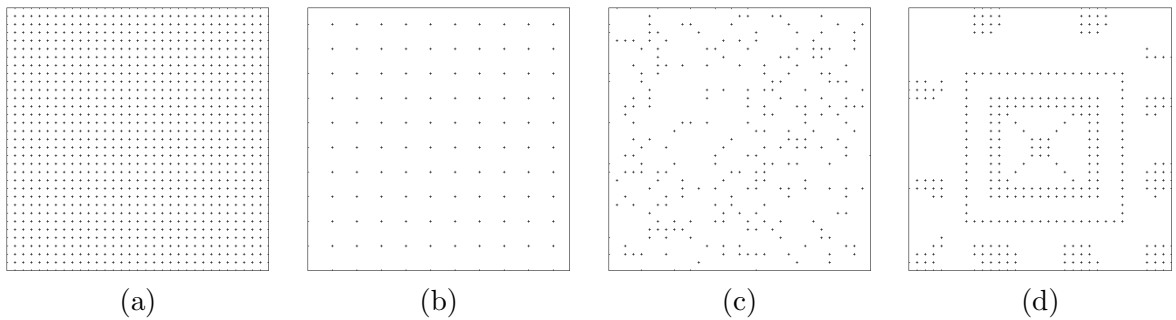


Figure 2: (a) A uniformly spaced grid of control points. (b) A uniformly spaced grid with a larger spacing between grid points. (c) A randomly spaced set of control points. (d) A set of control points with a varying density.

2 Thin-Plate Spline

Thin-plate spline (TPS) or surface spline [26, 40] is perhaps the most widely used transformation function in nonrigid registration. It was first used by Goshtasby [17] in the registration of remote sensing images and then by Bookstein [4] in the registration of medical images. Given a set of 3-D points as defined by (8), the TPS interpolating the points is defined by

$$f(x, y) = A_1 + A_2x + A_3y + \sum_{i=1}^N F_i r_i^2 \ln r_i^2, \quad (19)$$

where $r_i^2 = (x - x_i)^2 + (y - y_i)^2 + d^2$. This is the equation of a plate of infinite extent deforming under loads centered at $\{(x_i, y_i) : i = 1, \dots, N\}$. The plate deflects under the imposition of loads to take values $\{f_i : i = 1, \dots, N\}$. d^2 acts like a stiffness parameter. As d^2 approaches zero, the loads approach point loads. As d^2 increases, the loads become more widely distributed producing a smoother surface. Equation (19) contains $N + 3$ unknowns. By substituting the coordinates of N points as described by (8) into (19) and letting $f(x_i, y_i) = f_i$, N equations will be obtained. Three more equations are obtained using the following three constraints:

$$\sum_{i=1}^N F_i = 0, \quad (20)$$

$$\sum_{i=1}^N x_i F_i = 0, \quad (21)$$

$$\sum_{i=1}^N y_i F_i = 0. \quad (22)$$

Constraint (20) shows that the sum of the loads applied to the plate should be 0. This is needed to ensure that the plate would not move under the imposition of the loads and remain stationary. Constraints (21) and (22) require that moments with respect to x and y axes be zero, ensuring that the plate would not rotate under the imposition of the loads.

Using TPS to map Figs. 1b–h to Fig. 1a with the control points shown in Fig. 2, the results of Table 1 are obtained. The error measures in each entry of the table are obtained by using the coordinates of corresponding control points to determine the transformation parameters. The obtained transformation is then used to deform the target image to spatially align with the reference image. The control points shown in Figs. 2b–d were used to estimate the transformation functions, and the control points shown in Fig. 2a were used to estimate the maximum (MAX) error and the

Table 1: The registration accuracy of TPS.

		Trans.	Rot.	Sc.	Affine	Global	Local	G. & L.
Uniform, Dense	MAX	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	RMS	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Uniform, Sparse	MAX	0.0	1.4	1.4	1.4	20.0	27.0	33.5
	RMS	0.0	0.6	0.6	0.6	1.6	5.8	7.3
Random	MAX	0.0	1.4	1.4	1.4	33.0	14.1	33.0
	RMS	0.0	0.7	0.6	0.6	3.1	3.5	4.2
Varying Density	MAX	0.0	2.2	2.0	1.4	20.0	20.0	20.4
	RMS	0.0	0.8	0.7	0.7	1.6	6.7	6.7

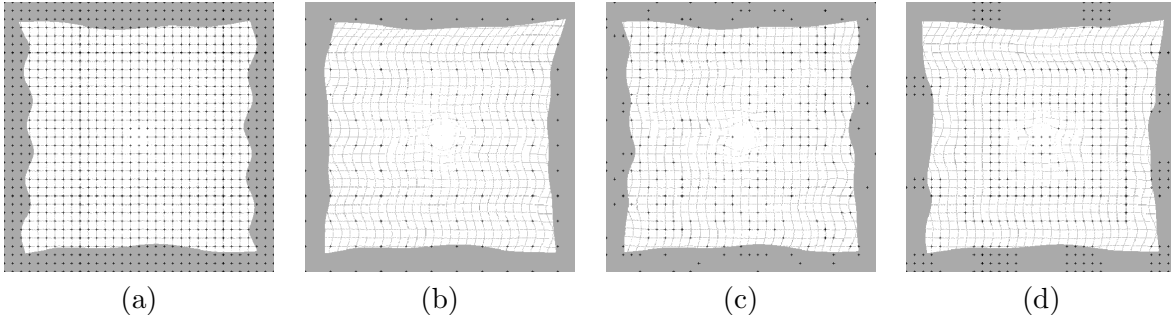


Figure 3: (a)–(d) The image shown in Fig. 1h after being resampled to align with the image shown in Fig. 1a with the control points shown in Figs. 2a–d, respectively.

root-mean-squared (RMS) error in each case. All errors in the first row in Table 1 are zero because TPS is an interpolating function and maps corresponding control points to each other exactly. Since the control points used to compute the transformation and the control points used to measure the error are the same, no errors are obtained.

In Table 1, the MAX registration errors for inverse distance, sinusoidal, and the combination of inverse distance and sinusoidal are much larger than the RMS errors for the same deformation because some grid points in the reference image fall outside the target image after warping and resampling. Since only grid points appearing inside both images are used to estimate the transformation, some of the test grid points near the image borders produce large errors, which contribute to the MAX error but contribute very little to the RMS error due to averaging effect. MAX error can be considered a local measure since its value reflects the error at a single grid point, while RMS error can be considered a global measure since its value reflects the errors at all the grid points.

Figures 3i–l show images obtained after resampling Fig. 1h to align with Fig. 1a using the control points shown in Figs. 2a–d, respectively. In these examples, although corresponding control points overlay perfectly, some errors are visibly large away from the control points.

When the images have translational differences, no registration errors are observed independent

of the density and organization of the points. This is because, first, coordinates of corresponding control points in the images do not contain rounding errors, and second, translation can be represented exactly by TPS due to its linear term. When the images have a rotational, scaling, or linear difference, TPS produces small errors that are primarily due to the rounding errors in the correspondences. Errors slightly increase when the density of control points varies across the image domain. When the images have nonlinear geometric differences, errors are rather large as shown in the rightmost three columns in Table 1.

From the results shown in Fig. 3 and Table 1, it can be concluded that although TPS performs relatively well when the images have global geometric differences, it performs rather poorly when the images have local geometric differences. This can be attributed to the fact that logarithmic basis functions are monotonically increasing and cover the entire image domain, thus, spreading a local deformation over the entire image domain. Also, logarithmic functions are radially symmetric and when the control points are irregularly spaced, large errors may be obtained away from the control points.

The stiffness parameter d^2 controls the shape of the interpolating surface and changes the registration accuracy. With the control point arrangements shown in Fig. 2, best accuracy was obtained when the stiffness parameter was zero. Larger stiffness parameters increase registration errors in these test cases. The errors reported in Table 1 are for a stiffness parameter of 0.

To reduce errors away from the control points, Rohr et al. [44, 55] added a smoothing term to the interpolating spline while $d^2 = 0$. As the smoothness parameter is increased, the obtained surface becomes smoother and fluctuations become smaller, but at the same time the surface gets farther from the control points. If interaction with the system is allowed, one may gradually change the smoothness parameter while observing the registration result and stop the process when the desired registration is achieved. When the control points are noisy and/or are very irregularly spaced, approximating splines are preferred over interpolating splines.

Table 2: Registration accuracy of MQ. Numbers inside parentheses show the errors when optimal d^2 was used, while numbers not in parentheses are the errors when $d^2 = 0$.

		Trans.	Rot.	Sc.	Affine	Global	Local	G. & L.
Uniform (Dense)	MAX	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	RMS	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Uniform (Sparse)	MAX	3.4	6.2	6.4	2.4	33.4	30.4	35.4
	RMS	1.1	1.3	1.4	1.0	4.8	5.0	6.4
Random	MAX	3.5	5.4	5.4	5.3	48.0	16.1	53.6
	RMS	1.0	1.3	1.1	1.2	5.0	5.3	7.4
Varying Density	MAX	4.2	6.8	6.5	20.9	56.4	33.2	71.8
	RMS	2.2	2.5	2.3	1.6	6.3	5.5	10.4
Uniform (Sparse)	MAX	(0.0)	(1.4)	(1.4)	(1.4)	(22.6)	(21.2)	(23.7)
	RMS	(0.0)	(0.6)	(0.6)	(0.6)	(4.4)	(4.5)	(4.8)
Random	MAX	(0.0)	(1.4)	(1.4)	(1.4)	(36.4)	(13.1)	(44.9)
	RMS	(0.0)	(0.6)	(0.6)	(0.6)	(2.7)	(2.8)	(3.3)
Varying Density	MAX	(0.0)	(2.0)	(2.0)	(1.4)	(12.4)	(18.3)	(48.9)
	RMS	(0.0)	(0.8)	(0.6)	(0.7)	(1.8)	(2.1)	(4.5)

3 Multiquadric

Multiquadric (MQ) interpolation [28, 29] is another radial basis function defined by

$$f(x, y) = F_1 + F_2x + F_3y + \sum_{i=4}^N F_i R_i(x, y), \quad (23)$$

where

$$R_i(x, y) = [(x - x_i)^2 + (y - y_i)^2 + d^2]^{\frac{1}{2}}. \quad (24)$$

Parameters $\{F_i : i = 1, \dots, N\}$ are determined by letting $f(x_i, y_i) = f_i$ for $i = 1, \dots, N$ and solving the obtained system of linear equations. As d^2 is increased, a smoother surface is obtained.

When registering Figs. 1b–h to Fig. 1a using MQ with the control points shown in Fig. 2, the results shown in Table 2 are obtained. Comparing these results with those in Table 1, it can be concluded that MQ and TPS produce similar results when the images do not have nonlinear geometric differences. This is because both formulations have a linear term. When the geometric difference between the images is global, TPS performs better than MQ. However, when the images have local geometric differences, TPS and MQ perform similarly, producing rather large errors. One point to note is that although TPS produces the best accuracy for the images tested in this paper for $d^2 = 0$, when MQ is used, $d^2 = 0$ does not produce the best accuracy in any of the cases. To determine the optimal d^2 a steepest descent algorithm is used, which makes MQ several times slower than TPS. The numbers in parentheses in Table 2 show the errors when the optimal parameter d^2 in each case is used to register the images.

4 Weighted Mean

TPS and MQ are interpolating functions. Therefore, they map corresponding control points in the images to each other exactly. Methods that map corresponding control points to each other approximately are obtained from a weighted average of the control points, with the sum of the weights equal to 1 everywhere in the approximation domain. A weighted mean (WM) method is represented by

$$f(x, y) = \sum_{i=1}^N f_i W_i(x, y) \quad (25)$$

where

$$W_i(x, y) = \frac{R_i(x, y)}{\sum_{i=1}^N R_i(x, y)} \quad (26)$$

is the i th weight function and $R_i(x, y)$ is a monotonically decreasing radial function centered at (x_i, y_i) .

When $R_i(x, y)$ is a Gaussian, the weights are called rational Gaussians [20]. Although Gaussians are symmetric, rational Gaussians are not symmetric because they have to ensure that the sum of the weights everywhere in the approximation domain is 1. This property makes the weight functions stretch toward the gaps.

As the widths of the weight functions are reduced, the obtained surface gets closer to the points. As the widths of the weight functions are reduced, flat spots start to appear in $f(x, y)$ surrounding the data points. This is demonstrated in a simple example in Fig. 4. Consider the following five data points: $(0, 0, 0)$, $(1, 0, 0)$, $(1, 1, 0)$, $(0, 1, 0)$, and $(0.5, 0.5, 0.5)$. The single-valued rational Gaussian surface approximating the points with increasing σ_i 's are shown in Figs. 4a–d. For small σ_i 's, flat horizontal spots are obtained around the data points. This means, for large variations in x and y , only small variations will be obtained in f near the data points. The implication of this is that when such a function is used to represent a component of a transformation function, points uniformly spaced in the reference image will map densely to points near the control points in the target image even when the reference and target images do not have any nonlinear geometric differences.

Consider the following 5 correspondences:

$$[(0, 0)(0, 0)]; [(1, 0)(1, 0)]; [(1, 1)(1, 1)]; [(0, 1)(0, 1)]; [(0.5, 0.5)(0.5, 0.5)]. \quad (27)$$

The transformation that maps uniformly spaced points in the reference image to points in the target image for increasing σ_i 's are depicted in Figs. 4e–h. The images show points in the target image when mapping to uniformly spaced points in the reference image. As σ_i 's are reduced, the density of surface points increases near the control points and the function more closely approximates the control points. As σ_i 's are increased, spacing between the points becomes more uniform, but the function moves away from the control points. Ideally, we want a transformation function that maps corresponding control points more closely to each other when σ_i 's are decreased without increasing the density of the surface points near the control points. Some variation in the density of the surface points is expected if the images to be registered have nonlinear geometric differences. Nonlinear geometric differences between two images change the local density of points in order to stretch some parts of the target image while shrinking the other parts to make its geometry resemble that of the reference image.

When the images do not have nonlinear geometric differences, in order to obtain uniformly spaced points in the target image for uniformly spaced points in the reference image, each component of the transformation should be formulated by a parametric surface. First, a parametric surface with components $x = x(u, v)$, $y = y(u, v)$, and $f = f(u, v)$ is computed fitting

$$\{(x_i, y_i, f_i) : i = 1, \dots, N\} \quad (28)$$

with parameter coordinates (nodes) $\{(u_i = x_i, v_i = y_i) : i = 1, \dots, N\}$. Given pixel coordinates (x, y) in the reference image, first, corresponding parameter coordinates (u, v) are determined from $x = x(u, v)$ and $y = y(u, v)$. Then, knowing (u, v) , $f = f(u, v)$ is evaluated. The obtained value will represent the X or the Y coordinate of the point corresponding to point (x, y) . Figs. 4i–l show the resampling results achieved in this manner. As can be observed, this transformation now maps uniformly spaced points in the reference image to almost uniformly spaced points in the target image while still mapping corresponding control points to each other with a reasonably good accuracy. The nonlinear equations are solved by initially letting $u = x$ and $v = y$ and iteratively

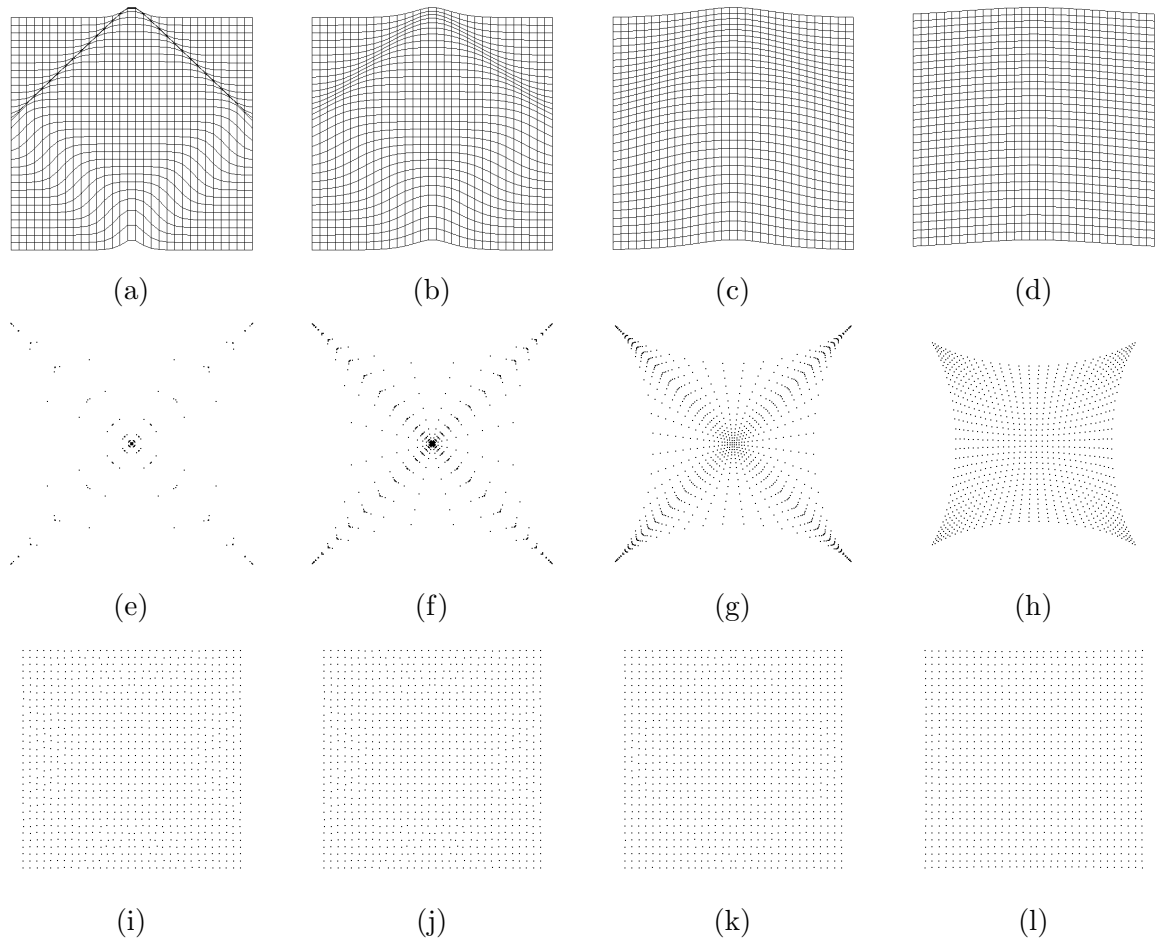


Figure 4: (a)–(d) A component of a transformation represented by a parametric rational Gaussian at increasing values of σ . (e)–(h) A transformation with components represented by single-valued rational Gaussian surfaces at increasing values of σ . (i)–(l) The same as in (e)–(h) but letting each component of the transformation be a parametric rational Gaussian surface.

revising u and v by Newton's method [34].

To determine the accuracy of WM in image registration, the control points shown in Fig. 2 and the deformations shown in Figs. 1 were used. MAX and RMS errors for rational Gaussian weights are shown in Table 3. Registration errors vary with the standard deviations of Gaussians, or the widths of the weight functions. As the standard deviations of Gaussians are increased, the transformation function becomes smoother and increase registration errors. As the standard deviations of Gaussians are reduced, the transformation function becomes more detailed, mapping corresponding control points to each other more closely. Too small a standard deviation, however, may cause larger errors away from the control points. There is a set of standard deviations that produces the least error for a given pair of images. In general, as variation in geometric difference between images becomes larger locally, smaller standard deviations are needed to reflect the sharper geometric differences.

The standard deviations of Gaussians are set inversely proportional to the density of the control points. This will take narrow Gaussians where the density of points is high and take wide Gaussians where density of points is low. The process can be made totally automatic by setting σ_i equal to the radius of the circular region surrounding the i th control point in the reference image containing $n - 1$ other control points ($n \geq 2$). n , which represents the neighborhood size, can be considered the smoothness parameter. The larger its value, the larger the neighborhood size and, thus, the smoother the obtained transformation will be. This is because as n is increased the widths of Gaussians increase, producing a smoother surface, and as n is decreased the widths of Gaussians decrease, producing a more detailed surface.

The results shown in parentheses in Table 3 were obtained when $n = 5$, and the results shown without parentheses were obtained when $n = 9$. When $n = 5$, MAX and RMS errors in Table 3 are smaller than those obtained for TPS and MQ as shown in Tables 1 and 2, except for errors in the first row. Since registration is done by approximation, corresponding control points do not map to each other exactly and so the errors are not zero. They are, however, very small. In all the experiments reported in this paper, only control points existing inside both images are used to calculate the transformation. Control points falling outside the target image after deformation

Table 3: Registration accuracy of the rational Gaussian. The errors are for $n = 9$. When reducing n to 5, the errors shown within parentheses are obtained.

		Trans.	Rot.	Sc.	Affine	Global	Local	G. & L.
Uniform (Dense)	MAX	0.0	1.0	0.0	1.0	1.8	2.2	2.5
	RMS	0.0	0.1	0.0	0.1	0.8	1.2	1.4
Uniform (Sparse)	MAX	0.0	1.4	1.0	1.4	32.6	9.0	35.6
	RMS	0.0	0.3	0.1	0.3	3.5	4.0	5.2
Random	MAX	0.0	1.4	1.4	1.4	36.0	10.2	36.0
	RMS	0.0	0.6	0.6	0.6	4.2	3.7	5.8
Varying Density	MAX	0.0	1.4	1.4	1.4	37.0	15.1	37.0
	RMS	0.0	0.5	0.6	0.5	3.5	4.0	6.6
Uniform (Dense)	MAX	(0.0)	(0.0)	(0.0)	(1.0)	(1.3)	(1.4)	(1.5)
	RMS	(0.0)	(0.0)	(0.0)	(0.1)	(0.4)	(0.6)	(0.8)
Uniform (Sparse)	MAX	(0.0)	(0.0)	(0.0)	(1.0)	(11.5)	(7.1)	(15.2)
	RMS	(0.0)	(0.0)	(0.0)	(0.1)	(1.4)	(1.2)	(1.5)
Random	MAX	(0.0)	(1.4)	(1.4)	(1.4)	(28.9)	(9.0)	(30.3)
	RMS	(0.0)	(0.4)	(0.4)	(0.3)	(2.8)	(1.3)	(3.0)
Varying Density	MAX	(0.0)	(1.4)	(1.4)	(1.4)	(31.4)	(15.0)	(33.2)
	RMS	(0.0)	(0.2)	(0.3)	(0.2)	(1.3)	(1.8)	(2.1)

and resampling of the reference image are not used in the calculations. Therefore, MAX errors in Tables 1–3 are rather large. Such large errors usually occur at a small number of grid points near the image borders.

When $n = 2$, the obtained function will be very detailed, reproducing noise in the correspondences. As n is increased, noise in the correspondences average out, producing a smoother transformation. Increasing n too much, however, will make it impossible to reproduce sharp geometric differences between the images. n should, therefore, be selected using information about noise in the correspondences as well as the density of the points.

Formula (25) defines a component of a transformation function in terms of a weighted average of a component of the control points in the target image. Computationally, this weighted mean approach is more stable than TPS and MQ because it does not require the solution of a system of equations. The process, therefore, can handle a very large number of control points with any density and organization. When TPS and MQ are used, the condition number of the matrix of coefficients increases with the size of the matrix of coefficients [7]. When density of control points varies greatly across the image domain, the system of equations to be solved may become ill-conditioned, producing large registration errors. This can be easily verified by moving even one of the control points sufficiently close to one other control point. A preprocessing step that removes some of the control points in very high density areas should improve the condition number of the matrix of coefficients in TPS and MQ.

Methods to make the weighted mean method local have been proposed by Maude [37] and

McLain [39] using rational weights defined by (26) but with

$$R_i(x, y) = \begin{cases} 1 - 3r_i^2 + 2r_i^3, & 0 \leq r_i \leq 1, \\ 0, & r_i > 1, \end{cases} \quad (29)$$

where

$$r_i = [(x - x_i)^2 + (y - y_i)^2]^{\frac{1}{2}} / r_n \quad (30)$$

and r_n represents the distance of control point (x_i, y_i) to its $(n - 1)$ st closest control point in the reference image. Note that since

$$\left[\frac{dR_i(x, y)}{dr_i} \right]_{r_i=1} = 0, \quad (31)$$

not only $R_i(x, y)$ vanishes at $r_i = 1$, it vanishes smoothly. Therefore, $f(x, y)$ will be continuous and smooth everywhere in the approximation domain. Knowing the weight function at a control point, the coefficients of a polynomial are then found to interpolate that point and $(n - 1)$ points closest to it. The weighted sum of the polynomials is then used to represent a component of the transformation. Therefore, the formula for a local weighted mean is

$$f(x, y) = \frac{\sum_{i=1}^N R_i(x, y) p_i(x, y)}{\sum_{i=1}^N R_i(x, y)} \quad (32)$$

where $p_i(x, y)$ is the polynomial interpolating data at (x_i, y_i) and those at $(n - 1)$ of its closest points. Note that this requires the solution of a small system of equations to find each local polynomial. The polynomials encode information about local geometric differences between the images. Therefore, the method is suitable for the registration of images where a large number of control points is available and sharp geometric differences exist between the images. When the density of control points does not vary greatly across the image domain, local polynomials of degree two will be sufficient. However, when density of control points varies greatly across the image domain, polynomials of degree two may create holes in the transformation where large gaps exist between the control points. Polynomials of higher degrees may be used to widen the local functions to cover the holes, but high-degree polynomials are known to produce fluctuations away from the data points. Therefore, when the density of control points varies greatly across the image domain, a globally defined weighted mean is more suitable for registering the images than a locally defined weighted mean.

5 Piecewise Linear

A piecewise linear (PL) transformation registers two images piece by piece, each piece using a linear transformation. Although so far only triangular regions have been used, in theory, the regions can be of any shape and size. PL mapping is continuous, but it is not smooth. When the regions are small or local geometric differences between images are small, PL may be sufficient. However, if local geometric differences between images are large, tangents at the two sides of a boundary shared by two triangles may become quite different, resulting in large registration errors.

If control points in the reference image are triangulated, by knowing the corresponding control points in the target image, the corresponding triangles in the target image will be known. The choice of triangulation will affect the registration accuracy. As a general rule, acute triangles should be avoided. Algorithms that maximize the minimum angle in triangles are known as Delaunay triangulation [25, 33]. A better approximation accuracy can be achieved if triangulation is carried out in 3-D using (x, y, f) coordinates as opposed to only (x, y) coordinates [3, 48].

In order to provide tangent continuity across triangles, polynomials of degrees two or higher are fitted to the triangles with the coefficients of the polynomials determined such that polynomial gradients at two sides of a triangle edge and in the direction normal to the edge become the same. Methods for fitting piecewise smooth surfaces to triangular meshes have been proposed [8, 10, 11, 46]. Such piecewise smooth surfaces may be used as the components of a transformation when registering images with local geometric differences. Another local transformation that assures tangent continuity across the image domain is the multi-level B-spline function proposed by Lee et al. [35] and Xie and Farin [57]. From a given set of corresponding control points, a hierarchy of uniform control lattices is created, representing a sequence of B-spline surfaces whose sum approaches f_i at (x_i, y_i) for $i = 1, \dots, N$.

Starting from 3-D triangle meshes, subdivision techniques can be used to create piecewise smooth surfaces over the triangles. Subdivision methods use corner-cutting rules to produce a limit smooth surface by recursively cutting off corners in the polyhedron obtained from the 3-D triangles [38, 41, 50]. Subdivision surfaces contain B-spline, piecewise Bézier, and non-uniform B-spline (NURBS) as special cases [47]. Therefore, each component of a transformation can be

Table 4: Registration accuracy of PL.

		Trans.	Rot.	Sc.	Affine	Global	Local	G. & L.
Uniform (Dense)	MAX	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	RMS	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Uniform (Sparse)	MAX	0.0	1.4	1.4	1.4	28.0	15.0	30.3
	RMS	0.0	0.5	0.5	0.5	3.4	4.0	4.6
Random	MAX	0.0	1.4	1.4	1.4	35.0	16.1	38.8
	RMS	0.0	0.6	0.6	0.6	4.0	4.0	4.1
Varying Density	MAX	0.0	1.4	1.4	1.4	37.0	16.0	40.1
	RMS	0.0	0.5	0.4	0.6	5.8	3.3	6.9

considered a B-spline, a piecewise Bézier, or a NURBS surface. These surfaces are very suitable for use as components of a transformation function for the registration of images with local geometric differences because they keep a local deformation or inaccuracy among the control point correspondences local.

PL transformation is efficient and often sufficient for nonrigid registration [18]. Use of piecewise cubic functions in nonrigid registration has also been explored [19]. Piecewise methods register image regions within the convex hull of the control points. Although extrapolation is possible outside the convex hull of the control points, the process could lead to large registration errors due to the very local nature of the function over each triangle.

Table 4 shows the results of mapping Figs. 1b–h to Fig. 1a with control points shown in Fig. 2 using PL transformation. Errors are slightly worse than those obtained by WM with rational Gaussian weights, but they are much better than those obtained by TPS and MQ. To obtain these results, the control points in the reference image were triangulated by Delaunay triangulation. Then corresponding triangles were obtained in the target image and a linear transformation was used to map a triangle in the target image to the corresponding triangle in the reference image.

6 Computational Complexity

Assuming the images being registered are of size $n \times n$ and N corresponding control points are available, the time needed to register the images consists of the time to compute the transformation function and the time to resample the target image to the coordinate system of the reference image. In the following, the computational complexities of the transformations discussed above are determined. An addition, a subtraction, a multiplication, a division, a comparison, or a small combination of them is considered an operation.

To determine a component of the TPS transformation, a system of N linear equations has to be solved. This requires on the order of N^2 operations. Resampling of each point in the target image into the space of the reference image requires use of all N correspondences. Therefore, resampling by TPS requires on the order of n^2N operations. Overall, the computational complexity of TPS is $O(N^2) + O(n^2N)$. The computational complexity of MQ is also $O(N^2) + O(n^2N)$. MQ is, however, several times slower than TPS as determination of the optimal parameter d^2 requires use of an iterative steepest descent algorithm.

WM transformation uses rational weights with coefficients that are the coordinates of the control points. Therefore, a transformation is immediately obtained from the coordinates of corresponding control points without solving a system of equations. Mapping of each point in the target image to the space of the reference image takes on the order of N operations because coordinates of all correspondences are used to find each resampled point. Therefore, overall, the computational complexity of the method is $O(n^2N)$. In practice, however, since monotonically decreasing functions, such as Gaussians, approach zero abruptly, it is sufficient to use only those control points that are in the immediate neighborhood of the point under consideration. Since digital accuracy is sufficient, use of control points farther than a certain distance to the point under consideration will not affect the result. For this to be possible though, the control points should be binned in a 2-D array so that given a point in the target image, the control points surrounding it could be determined without examining all the control points.

The computational complexity of PL transformation involves the time to triangulate the control points. This is on the order of $N \log N$ operations. Once the triangles are obtained, there is a need to find a transformation for each corresponding triangle. This requires on the order of N operations. Resampling of each point in the target image to the space of the reference image takes a small number of operations. Therefore, overall, the computational complexity of PL transformation is $O(N \log N) + O(n^2)$. For the images shown in Fig. 1 and the control points shown in Fig. 2, our implementations showed that PL was a few times faster than TPS, TPS was several times faster than MQ, and MQ was several times faster than WM.

Table 5 summarizes the computational complexities of TPS, MQ, WM, and PL transformations

Table 5: Computational complexities of various transformation functions.

Type of Transformation	Computational Complexity
TPS Interpolation	$O(N^2) + O(n^2N)$
MQ Interpolation	$O(N^2) + O(n^2N)$
WM Approximation	$O(n^2N)$
PL Interpolation	$O(N \log N) + O(n^2)$

when n and N are both very large.

7 Detecting the Incorrect Correspondences

Transformation functions contain information about the geometric differences between images. This information is sometimes crucial in understanding the contents of images. The presence of sharp differences in geometries of two images can be the result of local motion or deformation in the scene and may be significant in interpretation of the scene. Figs. 5a and 5b show the X and Y components of the true transformation for registering images in Figs. 1g and 1a. The sinusoidal geometric difference between the images is clearly reflected in the components of the transformation.

If some information about the geometric difference between two images is known, that information along with the obtained transformation can be used to identify the inaccurate correspondences. For instance, if the images are known to have only linear geometric differences, $f_x(x, y)$ and $f_y(x, y)$ will be planar. Planes are obtained only when all the correspondences are accurate though. Inaccuracies in the correspondences will result in small dents and bumps in the planes. The geometric difference between Figs. 1a and 1g is sinusoidal as demonstrated in Figs. 5a and 5b. This is observed when all the correspondences are accurate. In an experiment, two of the control points in Fig. 2a were displaced horizontally, two were displaced vertically, and one was displaced both horizontally and vertically, each by 15 pixels. The displacements are reflected in the obtained transformation as depicted in Figs. 5c and 5d. The dark and bright spots in these images are centered at the control points that were moved. A bright spot shows a positive displacement while a dark spot shows a negative displacement. When displacements are caused by inaccuracies in the correspondences, these small dents and bumps (dark and bright spots) identify the inaccurate correspondences.

Note that horizontal displacements appear only in the X -component of the transformation

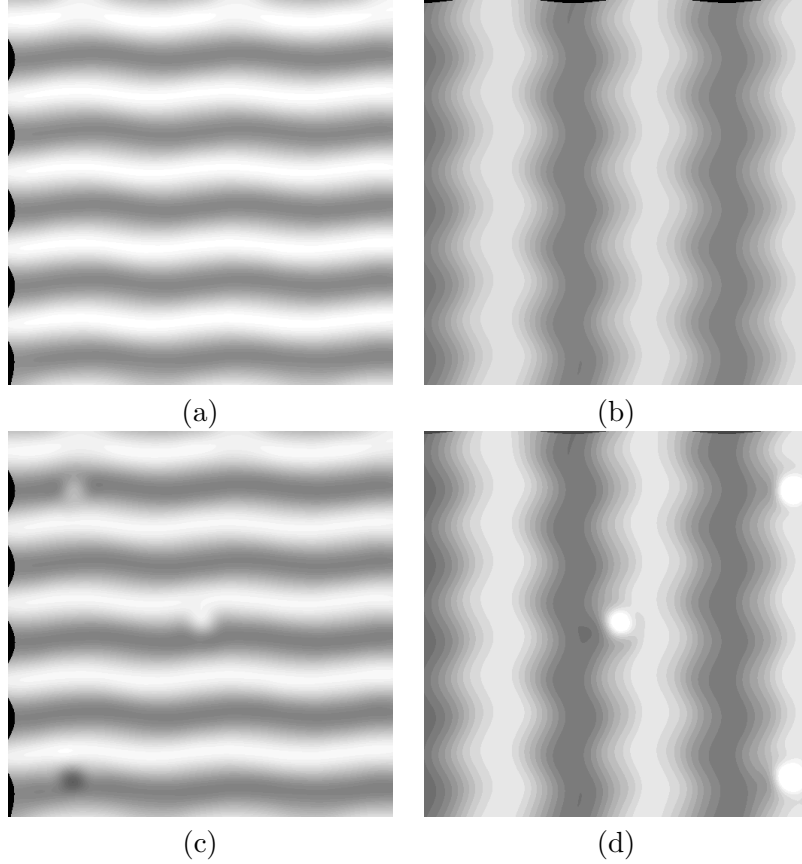


Figure 5: (a), (b) The X and Y components of the transformation mapping Fig. 1g to Fig. 1a. Shown here are $f_x(x, y) - x$ and $f_y(x, y) - y$ when mapped to 0–255 for enhanced viewing. Brighter points show higher functional values. (c), (d) The X and Y components of the transformation after displacing five of the control points in the target image.

and vertical displacements appear only in the Y -component of the transformation. Displacement in other directions will appear in both the X -component and the Y -component of the transformation. An inaccurate correspondence, therefore, could appear in one or both components of a transformation. A transformation contains valuable information about the accuracies of the correspondences, which can be used to identify and remove the inaccurate correspondences. If the geometric difference between two images is known to be gradual, large gradients in the components of a transformation will point to the inaccurate correspondences.

8 Conclusions

Images representing different views of a 3-D scene contain nonlinear geometric differences. To register such images, first a number of corresponding control points in the images must be determined

and then a transformation function that can accurately represent the geometric differences between the images must be found. Accuracy in image registration is, therefore, controlled by the accuracy of control-point correspondences and the accuracy of the transformation function that represents geometric differences between the images. This paper compared the accuracies of four popular transformation functions in nonrigid image registration.

If information about geometric differences between images is available, that information should be used to select the transformation. If no such information is available, a transformation function that can adapt to local geometric difference between the images should be chosen.

Use of TPS, MQ, WM, and PL transformations in nonrigid registration was explored. Experiments show that among the four transformations, TPS and MQ are least suitable for the registration of images with local geometric differences. This is because of three main factors. First, the basis functions from which a component of a transformation is determined are radially symmetric. When spacing between the control points is very irregular, large errors may be obtained in areas where large gaps exist between the control points. Second, because the radial basis functions used in TPS and MQ are monotonically increasing, the transformation becomes global and cannot adapt well to local geometric differences between images. Third, a system of equations has to be solved to find each component of a transformation, and when spacing between the control points varies greatly the system of equations to be solved may become ill-conditioned. When a small set of widely spread and accurate control points is given and local geometric differences between the images are not large, TPS and MQ are actually preferred over WM and PL because they are interpolating functions and map corresponding control points to each other exactly and also because they extend beyond the convex hull of the control points, registering entire images.

PL interpolation is most suitable when the images to be registered have local geometric differences because a control point affects registration in a small neighborhood of the control point. This property not only makes the computations very efficient, it keeps inaccuracies in the correspondences local without spreading them over the entire image domain. Recent subdivision schemes [38, 50] provide efficient means of fitting piecewise smooth surfaces to triangular meshes, creating continuous and smooth transformation functions.

When a large number of control point correspondences is given and the correspondences are noisy, WM is preferred over PL, TPS, and MQ for four main reasons. First, WM does not require the solution of a system of equations. A transformation is immediately obtained from the coordinates of corresponding control points in the images. Second, a transformation is obtained from a weighted average of the control points and the averaging process smoothes noise in the correspondences. Third, the rational weights adapt to the density and organization of the control points by automatically stretching toward large gaps and their widths change with the density of the control points. Fourth, the width of all weight functions can be increased or decreased together by controlling parameter n of the transformation.

Some applications require transformations that are spatially differentiable up to some degree. Among the four transformation functions discussed in this paper, TPS, MQ, and WM are C^∞ and PL is C^0 ; that is, all derivatives of TPS, MQ, and WM are continuous, while PL is continuous as a function, but its derivatives are all discontinuous across the image domain.

References

- [1] R. Bajcsy and S. Kovacic, "Multiresolution elastic matching," *Computer Vision, Graphics, and Image Processing*, vol. 46, pp. 1–21, 1989.
- [2] P. R. Beaudet, "Rotationally invariant image operators," *Proc. Int. Conf. Pattern Recognition*, pp. 579–583, 1978.
- [3] M. Bertram, J. C. Barnes, B. Hamann, K. I. Joy, H. Pottmann, and D. Wushour, "Piecewise optimal triangulation for the approximation of scattered data in the plane," *Computer Aided Geometric Design*, vol. 17, pp. 767–787, 2000.
- [4] F. L. Bookstein, "Principal warps: Thin-plate splines and the decomposition of deformations," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 11, no. 6, pp. 567–585, 1989.
- [5] G. Borgefors, "Hierarchical chamfer matching: A parametric edge matching technique," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 10, no. 6, pp. 849–865, 1988.

- [6] H. G. Borrow, J. M. Tenenbaum, R. C. Bolles, and H. C. Wolf, “Parametric correspondence and chamfer matching: Two new techniques for image matching,” *Proc. Joint Conf. Artificial Intelligence*, pp. 659–663, 1977.
- [7] R. E. Carlson and T. A. Foley, “The parameter R^2 in multiquadric interpolation,” *Computers Math. Applic.*, vol. 21, no. 9, pp. 29–42, 1991.
- [8] L. H. T. Chang and H. B. Said, “A C^2 triangular patch for the interpolation of functional scattered data,” *Computer-Aided Design*, vol. 29, no. 6, pp. 407–412, 1997.
- [9] L. P. Chew, M. T. Goodrich, D. P. Huttenlocher, K. Kedem, J. M. Kleinberg, and D. Kravets, “Geometric pattern matching under Euclidean motion,” *Computational Geometry Theory and Applications*, vol. 7, pp. 113–124, 1997.
- [10] C. K. Chui and M.-J. Lai, “Filling polygonal holes using C^1 cubic triangular spline patches,” *Computer Aided Geometric Design*, vol. 17, pp. 297–307, 2000.
- [11] P. Constantini and C. Manni, “On a class of polynomial triangular macro-elements,” *Computational and Applied Mathematics*, vol. 73, pp. 45–64, 1996.
- [12] A. D. J. Cross and E. R. Hancock, “Graph matching with a dual-step EM algorithm,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 20, no. 11 pp. 1236–1253, 1998.
- [13] W. A. Davis and S. K. Kenue, “Automatic selection of control points for the registration of digital images,” *Proc. Int’l J. Conf. Pattern Recognition*, pp. 936–938, 1978.
- [14] M. A. Fischler and R. C. Bolles, “Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography,” *Communications of the ACM*, vol. 24, pp. 381–395, 1981.
- [15] W. Förstner, “A feature based correspondence algorithm for image matching,” *Int. Arch. Photogramm. Remote Sensing*, vol. 26, pp. 150–166, 1986.
- [16] A. Goshtasby, “Template matching in rotated images,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 7, no. 3, pp. 338–344, 1985.

- [17] A. Goshtasby, "Registration of image with geometric distortion," *IEEE Trans. Geoscience and Remote Sensing*, vol. 26, no. 1, pp. 60–64, 1988.
- [18] A. Goshtasby, "Piecewise linear mapping functions for image registration," *Pattern Recognition*, vol. 19, no. 6, pp. 459–466, 1986.
- [19] A. Goshtasby, "Piecewise cubic mapping functions for image registration," *Pattern Recognition*, vol. 20, no. 5, pp. 525–533, 1987.
- [20] A. Goshtasby, "Design and recovery of 2-D and 3-D shapes using rational Gaussian curves and surfaces," *Int'l J. Computer Vision*, vol. 10, no. 3, pp. 233–256, 1995.
- [21] A. Goshtasby, S. Gage, and J. Bartholic, "A two-stage cross-correlation approach to template matching," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 6, no. 3, pp. 374–378, 1984.
- [22] A. Goshtasby and C. V. Page, "Image matching by a probabilistic relaxation labeling process," *Proc. 7th Int. Conf. Pattern Recognition*, vol. 1, pp. 307–309, 1984.
- [23] A. Goshtasby and G. Stockman, "Point pattern patching using convex hull edges," *IEEE Trans. Systems, Man, and Cybernetics*, vol. 15, pp. 631–637, 1985.
- [24] A. Goshtasby, G. Stockman, and C. Page, "A region-based approach to digital image registration with subpixel accuracy," *IEEE Trans. Geoscience and Remote Sensing*, vol. 24, no. 3, pp. 390–399, 1986.
- [25] P. J. Green and R. Sibson, "Computing Dirichlet tessellation in the plane," *Computer Journal*, vol. 21, pp. 168–173, 1978.
- [26] R. L. Harder and R. N. Desmarais, "Interpolation using surface splines," *J. Aircraft*, vol. 9, no. 2, pp. 189–191, 1972.
- [27] C. Harris and M. Stephens, "A combined corner and edge detector," *Proc. 4th Alvey Vision Conf.*, pp. 147–151, 1988.

- [28] R. L. Hardy, “Multiquadric equations of topography and other irregular surfaces,” *Journal of Geophysical Research*, vol. 76, no. 8, pp. 1905–1915, 1971.
- [29] R. L. Hardy, “Theory and applications of the multiquadric-biharmonic method - 20 years of discovery - 1969–1988,” *Computers Math. Applic.*, vol. 19, no. 8/9, pp. 163–208, 1990.
- [30] P. Huttenlocher and W. J. Rucklidge, “A multiresolution technique for comparing images using Hausdorff distance,” *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 705–706, 1993.
- [31] L. Kitchen and A. Rosenfeld, “Gray-level corner detection,” *Pattern Recognition Letters*, vol. 1, pp. 95–102, 1982.
- [32] D. Kozinska, O. J. Tretiak, J. Nissanov, and C. Ozturk, “Multidimensional alignment using the Euclidean distance transform,” *Graphical Models and Image Processing*, vol. 59, no. 6, pp. 373–387, 1997.
- [33] C. L. Lawson, “Software for C^1 surface interpolation,” in *Mathematical Software III*, J. R. Rice (Ed.), Academic Press, London, pp. 161–194, 1977.
- [34] J. L. Leader, *Numerical Analysis and Scientific Computation*, Addison-Wesley, New York, 2005.
- [35] S. Lee, G. Wolberg, and S. Y. Shin, “Scattered data interpolation with multilevel B-splines,” *IEEE Trans. Visualization and Computer Graphics*, vol. 3, no. 3, pp. 228–244, 1997.
- [36] H. Lester and S. R. Arridge, *Pattern Recognition*, vol. 32, no. 1, pp. 129–149, 1999.
- [37] A. D. Maude, “Interpolation—mainly for graph plotters,” *The Computer Journal*, vol. 16, no. 1, pp. 64–65, 1973.
- [38] J. Maillot and J. Stam, “A unified subdivision scheme for polygonal modeling,” *EURO-GRAPHICS*, A. Chalmers and T.-M. Rhyne (Eds.), vol. 20, no. 3, 2001.
- [39] D. H. McLain, “Two-dimensional interpolation from random data,” *The Computer Journal*, vol. 19, pp. 178–181, 1976.

- [40] J. Meinguet, “An intrinsic approach to multivariate spline interpolation at arbitrary points,” in *Polynomial and Spline Approximation*, B. N. Sahney (Ed.), D. Reidel Publishing Company, 163–190, 1979.
- [41] J. Peters and U. Reif, “The simplest subdivision scheme for smoothing polyhedra,” *ACM Trans. Graphics*, vol. 16, no. 4, pp. 420–431, 1997.
- [42] S. Ranade and A. Rosenfeld, “Point pattern matching by relaxation,” *Pattern Recognition*, vol. 12, pp. 269–275, 1980.
- [43] K. Rohr, “Localization properties of direct corner detectors,” *J. Math. Imaging and Vision*, vol. 4, pp. 139–150, 1994.
- [44] K. Rohr, H. S. Stiehl, R. Sprengel, T. M. Buzug, J. Weese, and M. H. Kuhn, “Landmark-based elastic registration using approximating thin-plate splines,” *IEEE Transactions on Medical Imaging*, vol. 20, no. 6, pp. 526–534, 2001.
- [45] J. Rucklidge, “Efficiently locating objects using the Hausdorff distance,” *Int. J. Computer Vision*, vol. 24, no. 3, pp. 251–270, 1997.
- [46] J. W. Schmidt, “Scattered data interpolation applying regional C^1 splines on refined triangulations,” *Math. Mech.*, vol. 80, no. 1, pp. 27–33, 2000.
- [47] P. Schröder and D. Zorin, “Subdivision for modeling and animation,” *SIGGRAPH Course No. 36 Notes*, 1998.
- [48] L. L. Schumaker, “Computing optimal triangulations using simulated annealing,” *Computer Aided Geometric Design*, vol. 10, pp. 329–345, 1993.
- [49] D. Shen and C. Davatzikos, “HAMMER: Hierarchical attribute matching mechanism for elastic registration,” *IEEE Workshop on Mathematical Methods in Biomedical Image Analysis*, Kauai, Hawaii, pp. 29–36, 2001.
- [50] J. Stam, “On subdivision schemes generalizing uniform B-spline surfaces of arbitrary degree,” *Computer Aided Geometric Design*, vol. 18, pp. 383–396, 2001.

- [51] G. Stockman, S. Kopstein, and S. Benett, “Matching images to models for registration and object detection via clustering,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 4, no. 3, pp. 229–241, 1982.
- [52] J-P Thirion and A. Gourdon, “Computing the differential characteristics of isointensity surfaces,” *Computer Vision and Image Understanding*, vol. 61, no. 2, pp. 190–202, 1995.
- [53] M. Trajković and M. Hedley, “Fast corner detection,” *Image and Vision Computing*, vol. 16, pp. 75–87, 1998.
- [54] G. J. Vanderburg and A. Rosenfeld, “Two-stage template matching,” *IEEE Trans. Computers*, vol. 26, no. 4, pp. 384–393, 1977.
- [55] G. Wahba, *Spline Models for Observation Data*, Philadelphia, PA, SIAM Press, 1990.
- [56] H. Wolfson and I. Rigoutos, Geometric hasing: An overview, *IEEE Computational Science & Engineering*, pp. 10–21, Oct./Dec. 1997.
- [57] Z. Xie and G. E. Farin, “Image registration using hierarchical B-splines,” *IEEE Trans. Visualization and Computer Graphics*, vol. 10, no. 1, 85–94, 2004.
- [58] S. Yam and L. S. Davis, “Image registration using generalized Hough transform,” *Proc. IEEE Conf. Pattern Recognition and Image Processing*, pp. 526–533, 1981.
- [59] C. T. Zhan, “An algorithm for noisy template matching,” *IFIP Congress*, Stockholm, pp. 698–701, 1974.