Image Warping with Scattered Data Interpolation Methods

Detlef Ruprecht and Heinrich Müller
Universität Dortmund
FB Informatik LS VII
Postfach 500 500
D-4600 Dortmund 50
Germany
phone: +49 - 231 - 755 6134
e-mail:
[ruprecht|mueller]@ls7.informatik.uni-dortmund.de

Forschungsbericht Nr. 443
6. November 1992

Abstract

Image warping has many applications in art as well as in image processing. Usually, displacements are computed with mathematical functions or by transformations of a triangulation of control points. Here, different approaches based on scattered data interpolation methods are presented. These methods provide smooth deformations with easily controllable behavior. The usefulness and performance of some selected classes of scattered data interpolation methods in this context is analyzed.

Zusammenfassung

Bildverzerrungen werden sowohl in künstlerischen Anwendungen als auch in der wissenschaftlichen Bildverarbeitung benötigt. Üblicherweise wird die Verschiebung der einzelnen Bildpunkte mit analytischen mathematischen Abbildungsfunktionen oder aus einer Triangulation der Kontrollpunkte gewonnen. In der vorliegenden Arbeit wird ein anderer Ansatz präsentiert, der auf Scattered–Data–Interpolationsmethoden basiert. Dieser Ansatz erzeugt glatte Deformationen mit einfach kontrollierbarem Verhalten. Die Brauchbarkeit und Effizienz verschiedener Scattered–Data–Methoden in diesem Zusammenhang wird verglichen.

Image Warping with Scattered Data Interpolation Methods

Detlef Ruprecht and Heinrich Müller

1 Digital Image Warping

The term "image warping" describes methods to deform images to arbitrary shapes. Digital image warping has received a lot of interest in recent years. Much of this interest is due to its large range of applications.

Some of them are in "artistic" areas, like computer animation. Here, image warping is a basis for two–dimensional morphing, the smooth transition between keyframe images. Other "artistic" applications include facial animation and free–form deformation of images. Another important area closely related to image warping is texture mapping, used for photorealistic rendering of surfaces in computer graphics [12].

Other applications are in scientific image processing. In image data acquisition, the acquisition method often causes deformations of the acquired image, e.g. through lens distortion. In satellite imaging, distortions are caused by surface curvature and oblique viewing angles. In ultrasonic imaging, distortions are caused by the varying speed of sound in different materials. These deformations have to be rectified to obtain correct coordinates. This process is called "registering". In registration applications, correspondence points between the deformed image and a reference image might be chosen manually or automatically. Considerable work has been done on automatically establishing correspondences, see e.g. [1, 9, 13, 20].

Another application related to registration is the normalization of samples to a standardized form to allow comparisons between the samples to be independent of size and slight form variations.

A medical application is the construction of voxel models from planar slices. The distance between slices is usually much larger than the spatial resolution within each slice. For rendering and for surface reconstruction, e.g. with the marching cube algorithm, however, it is preferable to have voxels with edges of equal length. To achieve this, some interpolation between slices is necessary. This is commonly done with linear interpolation between slices. This method is fast, but the result is not completely satisfactory. By deforming one slice towards the next, a better interpolation can be achieved.

The process of warping an image can be divided in two steps:

- The desired displacements of all pixels in the source image have to be computed.
- The image has to be resampled to create the output image.

The first step is usually done either by applying a global analytic mapping function to the image pixel positions, or by using a set of control points which specify the displacements of some points in the initial image. From these control points, a triangulation can be obtained and new locations computed by transforming each of the triangles [7, 8].

For the second step, a number of algorithms have been developed in recent years. A simple–minded approach would be to take the coordinates of four neighboring pixels, compute their deformed coordinates with the mapping function from above, and perform a bilinear fill for all pixels within the obtained rectangle ([5], p. 249). Much effort has been spent on more efficient algorithms for the resampling step. In particular, by separating the resampling step into horizontal and vertical displacements, efficiency can be improved considerably [19, 22]. For a detailed introduction to image warping methods, see [21].

This article focuses on the first step of image warping, the construction of a suitable mapping function. We will call this the “deformation” stage. It will be pointed out that triangulation based deformation is just a special case of scattered data interpolation methods applied to deformation. Triangulation based methods have the disadvantage of having non–continuous derivatives and not being defined outside the convex hull of the control points. For aesthetically pleasing results, a large number of triangles is needed. Other deformation methods are described which overcome these limitations. Their advantages and disadvantages are compared.

For the resampling stage, routines from the Khoros visualization system [16] are employed. These perform a bilinear fill as outlined above, and are therefore not very efficient. However, they are readily available in a user–friendly interactive image processing environment.

As an example, fig. 1b shows some subtle and some not so subtle deformations of the classic "Lenna" image, with the control point assignments shown in fig. 1a.

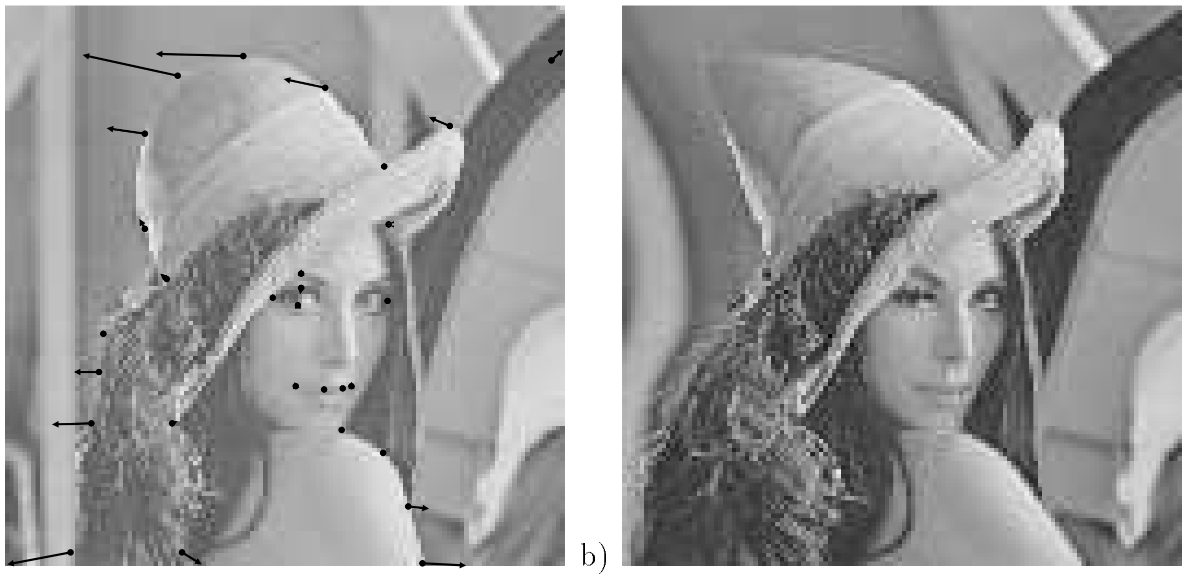

Figure 1: Deformations of a woman. a) A well known image from image processing papers, with control point assignments. b) The image deformed with multiquadrics.

The remainder of this paper is organized as follows. First, the relation between the problem of image deformation and the problem of scattered data interpolation methods is established. Several scattered data interpolation methods along with their application to image warping and discussion of their suitability follow. After that, the performance of the various methods is compared.

2 Image Deformation and Scattered Data Interpolation

The problem of image deformation can be formulated as follows:
Image Deformation Problem:
Input: n pairs $(\mathbf{p}_i, \mathbf{q}_i)$ of control points, $\mathbf{p}_i, \mathbf{q}_i \in \mathbb{R}^2$, $i = 1, \ldots, n$.
Output: An at least continuous function $\mathbf{f} : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ with $\mathbf{f}(\mathbf{p}_i) = \mathbf{q}_i$, $i = 1, \ldots, n$.

Besides continuity, other properties of the function $\mathbf{f}$ are desirable which concern the visual pleasantness and which are hard to describe in an exact manner. These properties will become recognizable in the following discussion of concrete propositions.

The problem of scattered data interpolation is to find a real valued multivariate function interpolating a finite set of irregularly located data points. For bivariate functions, this can be formulated as:

Scattered Data Interpolation Problem:
Input: n data points $(\mathbf{x}_i, y_i)$, $\mathbf{x}_i \in \mathbb{R}^2$, $y_i \in \mathbb{R}$, $i = 1, \ldots, n$.
Output: An at least continuous function $f : \mathbb{R}^2 \to \mathbb{R}$ interpolating the given data points, i.e. $f(\mathbf{x}_i) = y_i$, $i = 1, \ldots, n$.

As has been pointed out before [7, 19], the mapping functions can be split into their components $f_j : \mathbb{R}^2 \to \mathbb{R}$, $j = 1, 2$, and each component treated separately. These components can then be interpolated with scattered data interpolation methods. The data points to be used are $(\mathbf{p}_i, q_{i,j})$, $q_{i,j}$ the j-th component of $\mathbf{q}_i$.

3 Triangulation Based Methods

Scattered data interpolation through a triangulation of the data points is a classic approach in scientific visualization. This method consists of first dissecting the definition space into a suitable set of triangles with the given data points being the corners of the triangles. Then, each of the triangles is interpolated independently.

Several criteria for an "optimal" triangulation are known [15]. One optimal triangulation, which is based on a Dirichlet tessellation of the data set, is called "Delaunay triangulation". It can be computed with a divide–and–conquer algorithm of algorithmical complexity $O(n \log n)$ where n is the number of data points.

If the pixels lie on a regular grid, as is usually the case unless a previous deformation has already caused irregular pixel locations, the correspondence between triangles and pixels is trivially known in constant time, and thus each pixel can be mapped to its new location in constant time as well. As computing the new pixel location is the governing process in deformation, this gives an algorithmical complexity of $O(N)$ for triangulation based deformation, where N is the number of points to be mapped.

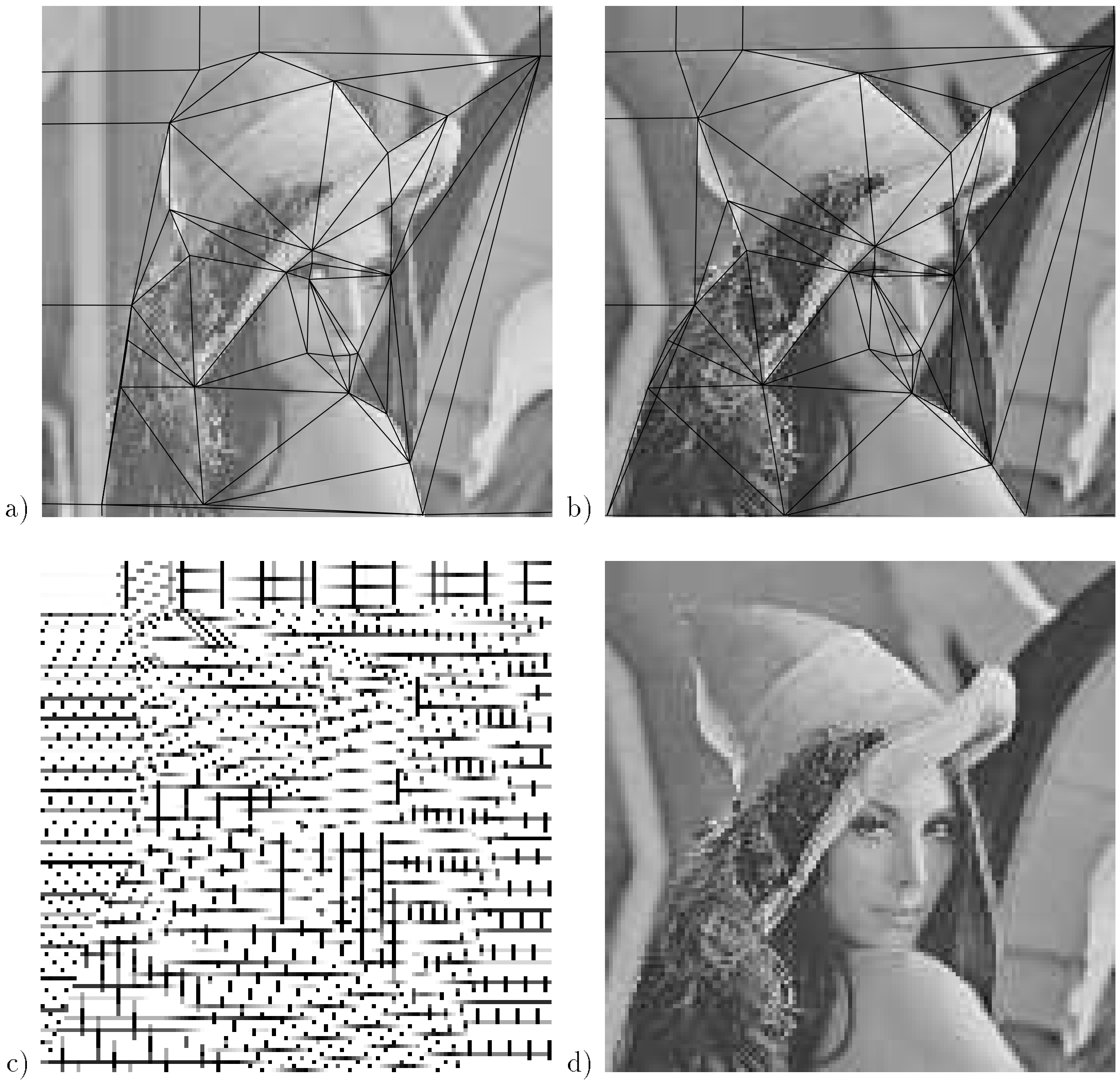

Figure 2: Deformation with triangulation based methods. a) Delaunay triangulation of the initial positions of all control points. b) Deformation of the triangulation. c) Deformed test grid with linear patches within the triangles. d) Deformed image with linear patches.

A continuous, although not smooth, interpolation can be obtained by linear interpolation using barycentric coordinates within each triangle. This method has been applied to image warping in [7]. The visual appearance of the result is quite acceptable if the deformations are small and if enough data points are provided so that changes of the transform coefficients between neighboring triangles are small. A smooth deformation can be obtained by using non–linear patches within the triangles.

In [8], a method is described which uses cubic functions obtained from a Clough–Tocher subtriangulation.

A problem common to all triangulation based methods for image warping is that foldover can easily occur. The term "foldover" describes the occurence of overlapping deformations, i.e. several non–adjacent pixels in the input image are mapped to the same pixel in the output image. With triangulation based methods, this happens if the orientation of the corner points changes for any of the triangles, i.e. the triangle is flipped over by the deformation.

Fig. 2 shows an example of deformation by triangulation with linear interpolation within the triangles. The Delaunay triangulation is shown along with the images. Note the edges in the deformed image where control points next to each other have been deformed dissimilarly. In the lower left corner, an example of foldover is observable.

4 Inverse Distance Weighted Interpolation Methods

Inverse distance weighted interpolation methods have been originally proposed by D. Shepard [18], and improved by a number of other authors, among them [3, 4, 6, 14]. For each data value i, a local interpolant $f_i(\mathbf{x}) : \mathbb{R}^2 \to \mathbb{R}$ with $f_i(\mathbf{x}_i) = y_i, i = 1, \ldots, n$ is determined. The interpolating function is a weighted average of these local interpolants, with weights dependent on the distance of the observed point from the given data points,

$$f(\mathbf{x}) \;=\; \sum_{i=1}^{n} w_i(\mathbf{x}) f_i(\mathbf{x}), \quad f_i(\mathbf{x}_i) = y_i, \; i = 1, \ldots, n. \tag{1}$$

$w_i : \mathbb{R}^2 \to \mathbb{R}$ is the weight function and $y_i \in \mathbb{R}$ the data value at the data point $\mathbf{x}_i \in \mathbb{R}^2$. The weight functions must satisfy the conditions

$$w_i(\mathbf{x}) \;\geq\; 0, \quad i = 1, \ldots, n, \quad \sum_{i=1}^{n} w_i(\mathbf{x}) = 1, \quad \text{and} \tag{2}$$

$$w_i(\mathbf{x}_j) \;=\; \delta_{ij}, \text{ i.e. } w_i(\mathbf{x}_i) = 1 \text{ and } w_i(\mathbf{x}_j) = 0, \; j \neq i, \; i, j = 1, ..., n. \tag{3}$$

These conditions guarantee the property of interpolation.

In [18], the following weight function was proposed:

$$w_i(\mathbf{x}) = \frac{\sigma_i(\mathbf{x})}{\sum_{j=1}^{n} \sigma_j(\mathbf{x})}\text{ , with} \tag{4}$$

$$\sigma_i(\mathbf{x}) = \frac{1}{(d_i(\mathbf{x}))^{\mu}}, \quad d_i(\mathbf{x}) \text{ the distance between } \mathbf{x} \text{ and } \mathbf{x}_i. \tag{5}$$

The smoothness is determined by the exponent μ. $\mu > 1$ assures continuity of the derivatives, with the value of the first derivative vanishing at the data points.

The original approach in [18] used just the value y_i at the data points for the local interpolant, i.e. $f_i(\mathbf{x}) = y_i$. This causes all points away from the data points to be moved towards the average of all y_i. While this might be tolerable for interpolation, it is unusable for image deformation, as it causes all points to be warped towards the center of the image, as demonstrated in fig. 3a. For this reason, linear or quadratic polynomials are normally used as local interpolants.

To determine the local interpolants, estimations of derivative values at the data points are needed. In [4], it has been proposed to compute the local interpolant $f_i(\mathbf{x})$ by minimizing the squared error of the mapping of other nearby control points $\mathbf{x}_j$ with f_i. In our experiments, we utilize all other control points, weighted with a coefficient w_{ij} to attenuate the influence of distant points. The corresponding error function $E_i(f)$ is

$$E_i(f) = \sum_{j=1, j\neq i}^{n} w_{ij} \cdot \|f(\mathbf{x}_j) - y_j\|^2. \tag{6}$$

The optimum local interpolant is obtained by minimizing $E_i(f)$. The coefficients w_{ij} should depend on the distance between $\mathbf{x}_i$ and $\mathbf{x}_j$ in a manner similar to $w_i(\mathbf{x})$, but it is not possible to use w_i because of (3). A simple possibility is to use $w_{ij} = \sigma_i(\mathbf{x}_j)$. This approach has been used in our experiments with local interpolants.

The application of inverse distance weighted interpolation to image warping gives

$$\mathbf{f}(\mathbf{p}) = \sum_{i=1}^{n} w_i(\mathbf{p})\mathbf{f}_i(\mathbf{p}), \quad \mathbf{f}_i(\mathbf{p}_i) = \mathbf{q}_i, \ i = 1, \ldots, n. \tag{7}$$

$\mathbf{f}_i : \mathbb{R}^2 \to \mathbb{R}^2$ are the local interpolants. They can be rewritten as

$$\mathbf{f}_i(\mathbf{p}) = \mathbf{q}_i + \mathbf{D}_i(\mathbf{p} - \mathbf{p}_i), \text{ with } \mathbf{D}_i : \mathbb{R}^2 \to \mathbb{R}^2,\ \mathbf{D}_i(0) = 0. \tag{8}$$

A simple possibility is to choose the $\mathbf{D}_i$ as linear transformations. Then $\mathbf{D}_i$ can be represented by a 2×2–matrix $(d_{i,kl})$, $k,l = 1,2$. The error function corresponding to a linear transformation $\mathbf{D}$ is

$$\begin{aligned} E_i(\mathbf{D}) &= \sum_{j=1,j\neq i}^{n} w_{ij} \cdot \left\| \mathbf{q}_i + \begin{pmatrix} d_{11} & d_{12} \\ d_{21} & d_{22} \end{pmatrix} (\mathbf{p}_j - \mathbf{p}_i) - \mathbf{q}_j \right\|^2 && (9) \\ &= \sum_{j=1,j\neq i}^{N} w_{ij} [(d_{11}(p_{j,1} - p_{i,1}) + d_{12}(p_{j,2} - p_{i,2}) + q_{i,1} - q_{j,1})^2 + \\ &\quad +(d_{21}(p_{j,1} - p_{i,1}) + d_{22}(p_{j,1} - p_{i,1}) + q_{i,2} - q_{j,2})^2]. && (10) \end{aligned}$$

The minimum of the error function is obtained with the partial derivatives with respect to the d_{kl}, $k,l = 1,2$. These are linear functions of the d_{kl}. Thus, we obtain a system of four linear equations with four unknowns, which can easily be solved for the d_{kl}. For a detailed description of deformation with inverse distance weighted methods, see [17].

With a linear local interpolant, the deformation has linear precision, i.e. if the requested displacements of the control points can be satisfied by a linear transformation, this is what will be obtained.

An advantage of inverse distance weighted methods is the large number of ways they can be tailored to specific needs. The range of influence of the control points can be limited by weight functions which vanish outside a distance R [4]. The interpolation condition can be relaxed by adding a damping term to the weight function, or strict adherence to the local interpolant can be enforced within a distance R around control points [17]. Many parameters can be tweaked to control the behavior of the deformation.

Fig. 3 shows examples of image deformations through inverse distance weighted methods with linear local interpolants and various values for μ. The results are clearly not too satisfactory. The main reason is that the transition between the range of influence of the control points is rather uneven. This causes unwanted bends in the deformation, as can be seen in the shoulder and arm area of the picture. Experiments with other weight functions and other exponents μ have not led to significant improvements. The best results were obtained with locally bounded weight functions [4], fig. 3d, but even this image shows some unevenness, and the quality depends strongly on the choice of parameters. Quadratic local interpolants might yield better results, but the method is already very

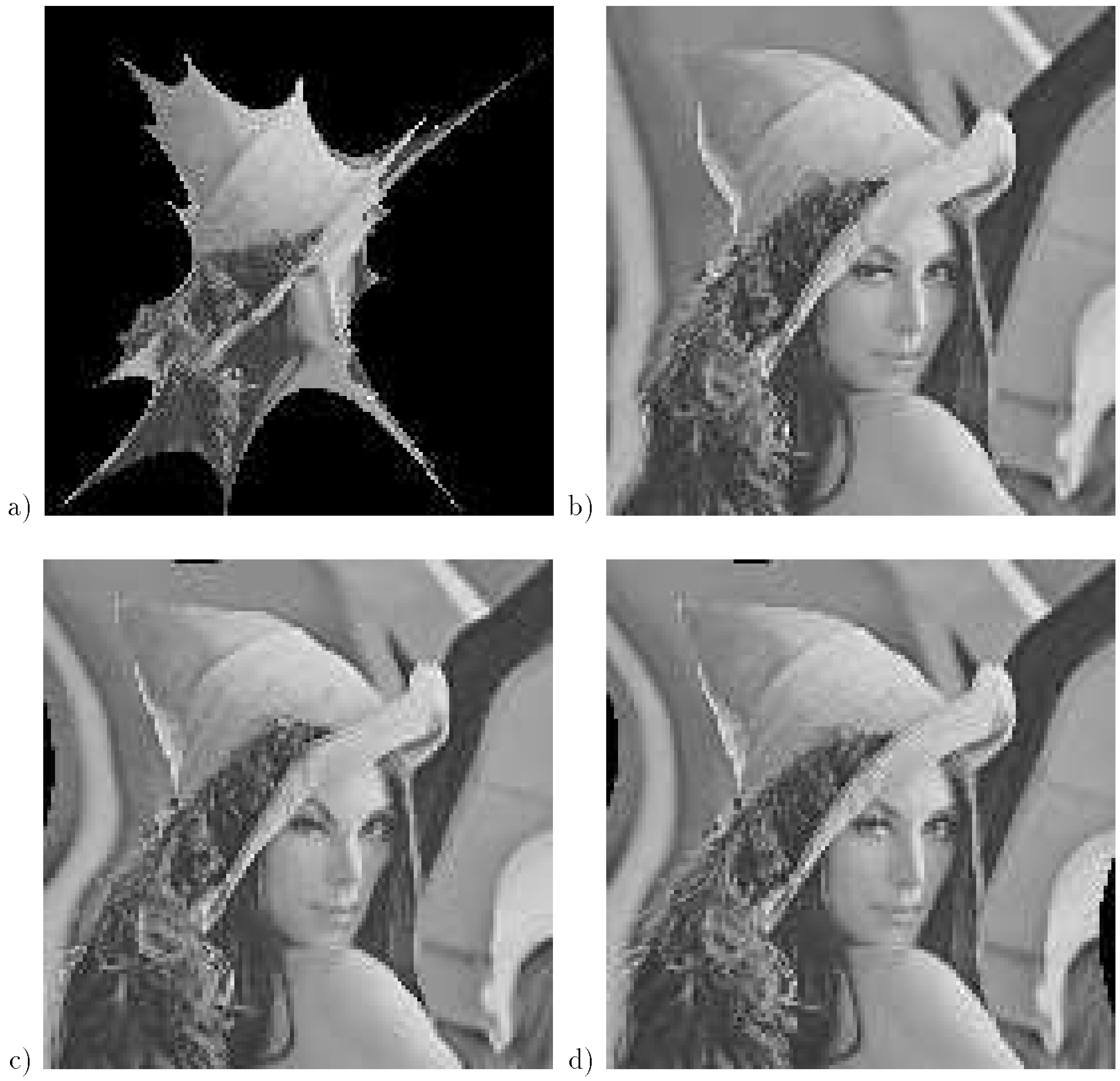

Figure 3: Deformation with inverse distance weighted methods. a) Original Shepard approach without local interpolant. b) Linear local interpolant, $\mu = 2$. c) Linear local interpolant, $\mu = 4$. d) Locally bounded weight functions, $R = 256$.

time–consuming with linear interpolants. With quadratic interpolants, this drawback would become more severe. Thus, even though the method is flexible, it does not appear very suitable for image warping.

5 Radial Basis Functions

Another popular approach to scattered data interpolation is to construct the interpolant as a linear combination of basis functions and then to determine the coefficients of the basis functions,

$$f(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i \; R(d_i(\mathbf{x})) + p_m(\mathbf{x}). \tag{11}$$

The values of the basis function R depend only on the distance from the data point, and are thus called radial. $p_m(\mathbf{x})$ is a polynomial of degree m. It assures a certain degree m of polynomial precision. The coefficients α_i are calculated by putting the data points into (11) and solving the resulting system of linear equations. The differentiability of this interpolation method depends directly on the differentiability of the basis functions R used.

Well–known radial basis functions are multiquadrics, originally proposed by R. Hardy [10, 11],

$$R(d) = (d^2 + r^2)^{\mu/2} \quad \text{with} \quad r > 0 \text{ and } \mu \neq 0. \tag{12}$$

Hardy's original approach does not contain a polynomial p_m. Therefore, it has a polynomial precision of 0. For the exponent μ, Hardy proposes $\mu = 1$. $\mu = -1$ has also been used successfully. For $r \neq 0$, the basis functions are infinitely differentiable, so the resulting interpolation is also in C^∞.

The characteristic radius $r > 0$ can be chosen arbitrarily. It determines the smoothness of interpolation at the given data points. For image deformation, the selection of r is critical for good results. Values too small lead to undesirable unevenness in the deformed image, values too big can lead to foldover. With fixed values of r, both effects are likely to occur even within the same image, as shown in fig. 4a, where foldover is evident near the eye and, at the same time, the rim of the hat is unpleasantly bent. Thus, it appears necessary to choose individual values of r for each control point.

Following a suggestion in [2], we use individual values r_i for each data point $\mathbf{p}_i$, computed from the distance to the nearest neighbor, i.e.

$$r_i = \min_{i \neq j} \; d_i(\mathbf{x}_j). \tag{13}$$

This causes the deformation to be softer when data points are widely spaced and stronger when they are closer together.

The application of radial basis functions to the problem of deformation gives

$$\mathbf{f}(\mathbf{p}) = \sum_{i=1}^{n} \boldsymbol{\alpha}_i \; R(d_i(\mathbf{p})) + \mathbf{p}_m(\mathbf{p}). \tag{14}$$

Now $\boldsymbol{\alpha}_i \in \mathbb{R}^2$ are coefficient vectors and $\mathbf{p}_m : \mathbb{R}^2 \to \mathbb{R}^2$ is a function with polynomial components of degree m.

It turns out that Hardy's multiquadrics approach without a polynomial term $\mathbf{p}_m$ does not yield satisfactory results as it produces distortions even if all control points remain at their original positions and diverges quickly outside the convex hull of the control points. This behavior is caused by the non–linearity of the basis functions. Introducing a linear $\mathbf{p}_m$, i.e. $m = 1$, as a global transformation with coefficients computed by the least squares method to minimize errors for the known control points solves this problem. More easily, under most circumstances, an identical transform is sufficient. The resulting mapping function is then

$$\mathbf{f}(\mathbf{p}) = \sum_{i=1}^{n} \boldsymbol{\alpha}_i \; ((d_i(\mathbf{p}))^2 + r_i^2)^{\mu/2} + \mathbf{p} \tag{15}$$

The image in fig. 4d has been computed with this method, with $\mu = 1$. Fig. 4c shows a test grid deformed with the same parameters to demonstrate the smoothness of the deformation.

A disadvantage of radial basis functions – like all global interpolation methods – is that for each pixel, all control points have to be taken into account. Thus, the algorithmic complexity is $O(n \cdot N)$, where n is the number of control points and N the number of pixels to be displaced, as opposed to triangulation based methods with a complexity $O(N)$. There is, however, room for improvements in computational efficiency in radial basis functions.

For radial basis functions with $\mu = \pm 1$, $n \cdot N$ square roots have to be computed. This is undesirable even if a fast square root approximation is used. Fortunately, our experiments show that $\mu = -2$ gives results which are almost as good, as shown in fig. 4b. With this exponent, the square root is replaced by a division.

Additionally, if the pixel locations of the input image are on a regular grid, and the initial locations of the control points also lie on that grid, the number of distances that has to be evaluated can be greatly reduced by computing the pixel displacements per control point radially outward from the control point, so that the typically eight grid points with the same distance from the control point are treated simultaneously. The performance gain from this algorithm as compared to a straightforward scanline algorithm is, however, largely offset by the larger number of boundary checks necessary.

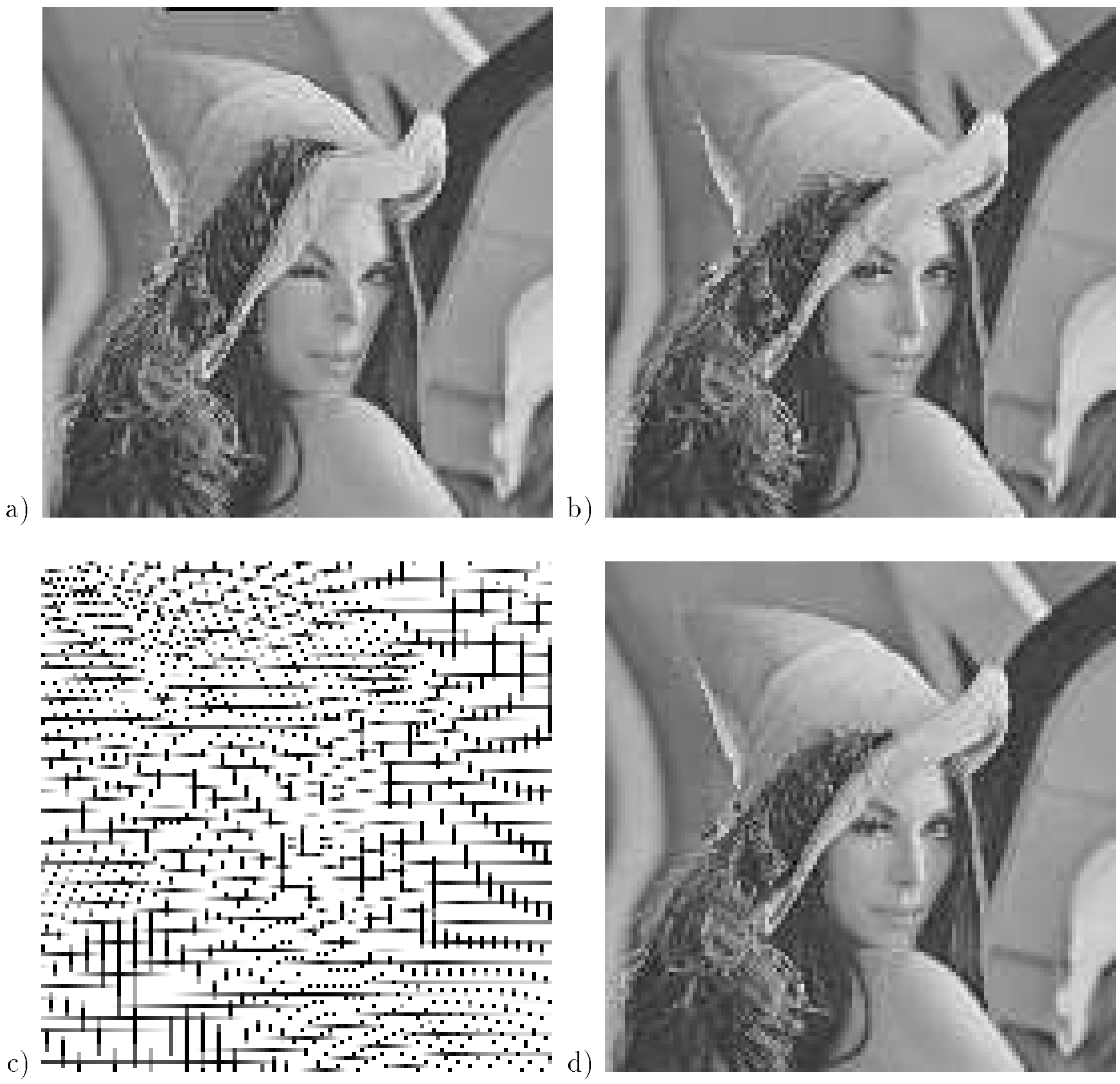

Figure 4: Deformation with radial basis functions. a) Fixed $r = 50$, $\mu = 1$. b) Adaptive r, $\mu = -2$. c) Test grid deformed with adaptive r, $\mu = 1$. d) Image deformed with the same parameters.

6 Locally Bounded Radial Basis Functions

The algorithm outlined above, where evaluation of pixel displacements spreads outwards from the control points, also opens the door for another more significant improvement.

As pointed out above, multiquadrics with a value of $\mu = -2$ give quite reasonable results for image deformation. These vanish for large distances. It therefore seems plausible to limit the range of influence of the basis functions, so that evaluation stops at some distance from the control points. This way it is possible to beat the $O(n \cdot N)$ complexity limit.

The range of influence can be limited by multiplication with a mollifying function, so that the radial basis function has the form

$$R_i(d) = (d^2 + r_i^2)^\mu \cdot (1 - (d/L_i)^2)_+^\nu \qquad \text{with} \qquad x_+^\nu = \left\{ \begin{array}{ll} x^\nu, & \text{if } x > 0 \\ 0, & \text{if } x \leq 0 \end{array} \right. \tag{16}$$

where L_i is the limit of the range of influence of the control point $\mathbf{p}_i$. The resulting basis functions are in $C^{\nu-1}$. In our experiments, a simpler and more efficient approach has been used. We subtract a constant offset from the basis functions, i.e.

$$R_i(d) = \left((d^2 + r_i^2)^\mu - \delta_i\right)_+ \tag{17}$$

where δ_i is chosen so that $R_i(L_i) = 0$. The derivatives of this function are not defined for $d = L_i$, so that $R_i \in C^0$, but for sufficiently large L_i, this does not matter in practice.

Two strategies for determining the L_i have been tried.

The first strategy just sets L_i to a multiple of r_i, the characteristic radius of the multiquadrics. This ensures some overlap of the influence ranges of near neighbors. However, it can lead to gaps where no deformation takes place if control points are spaced very irregularly.

The second strategy sets L_i to a distance where the influence of R_i is below a certain threshold T. To achieve this, the coefficients α_i are first evaluated for normal multiquadrics (eq. 12). The offset δ_i is then set to $\delta_i = T/\max(|\alpha_{i,x}|, |\alpha_{i,y}|)$. This way, the influence of the modified radial basis function vanishes where the original multiquadric would have had an influence of at most T. The coefficients α_i then have to be re–evaluated for the modified multiquadrics with the offset δ_i. This could be iterated

a) b)

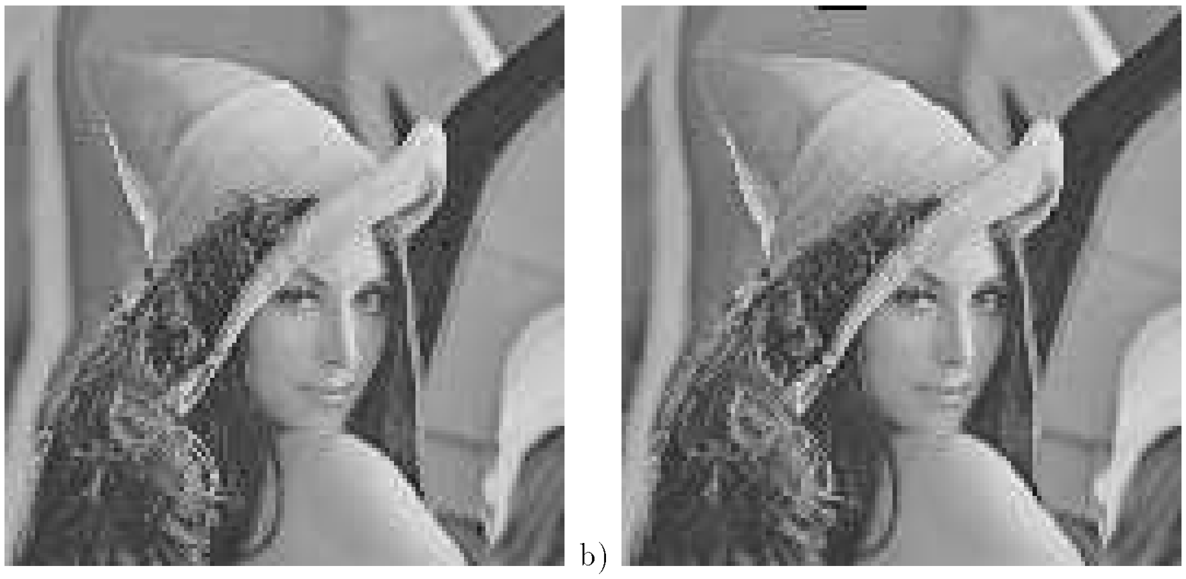

Figure 5: Deformation with locally bounded radial basis functions. a) Locally bounded with $L_i \geq 2r_i$ and $T = 5$. b) Locally bounded with $L_i \geq 1.4r_i$ and $T = 10$.

several times, but from our experience, one iteration step appears to be quite sufficient.

The second strategy has one major flaw in that it is possible to get $L_i = 0$, which would cause the linear equation system for the α_i to have no solution. This problem is solved by specifying a minimum range L_{min} so that $L_i \geq L_{min}$. For L_{min}, we use a value determined with the first strategy. This combination of the two strategies also helps to obtain greater stability with regards to unwanted artefacts in the deformation, like foldover or local roughness.

Our experiments show that with locally bounded radial basis functions, speedups in the order of 10 can be achieved without serious loss of quality.

Fig. 5 shows examples of locally bounded multiquadrics. Fig. 5a shows the result with a threshold of $T = 5$ pixels and a minimum range of $2r_i$. There is no visible difference to multiquadrics with $\mu = -2$, but the computation was about seven times faster. Fig. 5b shows deformation with $T = 10$ and a minimum range of $1.4r_i$. Despite the fact that the time required for the deformation is halved again, the quality is quite striking. The only apparent problem is the slight foldover near the tip of the hat.

7 Performance Comparisons

Table 1 shows the result of some run–time measurements with the example picture used throughout this article. This is a 512×512 pixel, 256 gray levels image, deformed with 29 control points. The measurements took place on a Sun Sparc2. The measured time value is the user time spent on behalf of the deformation, as determined from calls to the system function `rusage` before and after the call to the deformation routine. This time measurement is relatively unaffected by system load. No file operations take place during the measurement.

Detailed profiling shows that very little of the time needed for deformation is spent for preparational operations like solving the equation system for multiquadrics. Almost all the time is used for the actual evaluation of the transformed locations of the pixels.

deformation method	average time
triangulation, corners	0.9 sec
triangulation, poles	1.6 sec
inverse distance, $\mu = 2$	75.6 sec
multiquadrics, scanline, $\mu = 1$	92.4 sec
multiquadrics, scanline, $\mu = -2$	29.4 sec
multiquadrics, radial, $\mu = -2$	24.8 sec
multiquadrics, bound $= 3$	4.9 sec
multiquadrics, bound $= 2$, thresh $= 5$	3.4 sec
multiquadrics, bound $= 1.4$, thresh $= 10$	1.7 sec
resampling with bilinear fill	32.1 sec

Table 1: Run–time measurements for the various algorithms.

For triangulation based methods, four more control points have to be added to ensure that the convex hull of the control points covers the whole image. In the "corners" measurement, these points were in the corners of the image. In the "poles" measurement, the points were positioned far away in the north, east, west, and south directions of the image. This ensures that the interior triangulation is not affected by the added points, but adds some overhead for points outside the image. This second set of additional points was used for the images shown.

For inverse distance weighted methods with $\mu = 2$, no exponentiation is necessary. Other exponents give lower performance.

For multiquadrics with $\mu = 1$, the square root function from the standard math library is used. For, $\mu = -2$, no exponentiation is necessary. The result of the deformation with locally bounded radial basis functions with fixed $L_i = 3\,r_i$ is very similar to that shown in fig. 5a, but takes more time to compute.

There is a number of possibilities to further increase the speed of computation. One approach would be to reduce the number of points mapped with a global method by skipping pixels and mapping the pixels in between with a simpler method, e.g. a bilinear interpolation.

Also, if displacements are computed with a scanline algorithm, the operation to be executed is the same for all pixels, so that global interpolation methods, and radial basis functions in particular, are ideally suited for parallelization.

In an interactive environment, locally bounded multiquadrics could be further modified so that when a new control point is added, only the coefficient for this new control point is evaluated to displace this point as desired, and the complete system of equations re–evaluated only when time permits. The displacements caused by a single point can be evaluated in a small fraction of a second.

However, with the current implementation of the resampling process, this by far dominates the overall time needed for warping, so that further improvements of the performance of the deformation stage do not appear useful at the moment. The time needed for resampling the test image is given in the last row of table 1.

8 Conclusion

Mapping functions for image warping based on scattered data interpolation methods have been described. Methods based on radial basis functions, particularly multiquadrics, have been shown to be suitable for application to image warping. These functions produce deformations in C^∞, and are relatively insensitive to foldover problems. For strong deformations, where foldover is more likely to occur, the method lends itself well to decomposition into step–wise deformation [17].

A disadvantage of global deformation methods relative to the more common triangulation based methods is the higher algorithmical complexity. A local method based on multiquadrics has been presented that overcomes this problem without sacrificing quality. The performance of this method can compete with that of triangulation based methods and is likely to surpass them if smooth interpolation is used within the triangles.

Further work will be devoted to the application of these methods to two–dimensional morphing and to the interpolation between planar slices in medical data sets. Also, since the methods described here depend only on Euclidean distance, they can easily be adapted to volume deformation applications, e.g. the registration of voxel data.

References

[1] P. Buche and J. Camillerapp. Serial cuttings matching: An application to muscle fiber characterization. In *Proc. EUROGRAPHICS '91*, pages 329–340, 1991.

[2] M. Eck. Interpolationsmethoden zur Rekonstruktion von 3D–Oberflächen aus ebenen Schnittfolgen. *CAD und Computergraphik*, 13(5):109–120, Feb. 1991.

[3] R. Farwig. Rate of convergence of Shepard's global interpolation formula. *Mathematics of Computation*, 46(174):577–590, 1986.

[4] R. Franke and G. Nielsen. Smooth interpolation of large sets of scattered data. *Int. Journal for Numerical Methods in Engineering*, 15:1691–1704, 1980.

[5] R.C. Gonzalez and P. Wintz. *Digital Image Processing.* Addison Wesley, Reading, MA, 2nd edition, 1987.

[6] W.J. Gordon and J.A. Wixom. Shepard's method of "metric interpolation" to bivariate and multivariate interpolation. *Mathematics of Computation*, 32(141):253–264, 1978.

[7] A. Goshtasby. Piecewise linear mapping functions for image registration. *Pattern Recognition*, 19(6):459–466, 1986.

[8] A. Goshtasby. Piecewise cubic mapping functions for image registration. *Pattern Recognition*, 20(5):525–533, 1987.

[9] A. Goshtasby, G.C. Stockman, and C.V. Page. A region–based approach to control point selection with subpixel accuracy. *IEEE Trans. Geosci. Remote Sens.*, GE-24:390–399, 1986.

[10] R.L. Hardy. Multiquadric equations of topography and other irregular surfaces. *J. Geophys. Res.*, 76:1905–1915, 1971.

[11] R.L. Hardy. Theory and applications of the multiquadric–biharmonic method. *Computers and Mathematics with Applications*, 19:163–208, 1990.

[12] P.S. Heckbert. Survey of texture mapping. *IEEE Computer Graphics and Applications*, 6(11):56–67, 1986.

[13] L.N. Kanal, B.A. Lambird, D. Levine, and G.C. Stockman. Digital registration of images from similar and dissimilar sensors. In *Proc. Int. Conf. Cybern. Society*, pages 347–351, 1981.

[14] P. Lancaster and K. Salkauskas. Surfaces generated by moving least squares methods. *Mathematics of Computation*, 37(155):141–158, 1981.

[15] C.L. Lawson. Software for C^1 surface interpolation. In J.R. Rice, editor, *Mathematical Software III*, pages 161–194. Academic Press, 1977.

[16] J. Rasure and M. Young. An open environment for image processing software development. In *1992 SPIE/IS&T Symposium on Electronic Imaging, SPIE Proceedings Vol. 1659*, February 1992.

[17] D. Ruprecht and H. Müller. Free form deformation with scattered data interpolation methods. *Universität Freiburg, Institut für Informatik, Bericht 41*, 1991. To be published in *Computing Suppl. 8*, Springer–Verlag, Wien 1992.

[18] D. Shepard. A two–dimensional interpolation function for irregularly spaced data. In *Proc. 23 Nat. Conf. ACM*, pages 517–524, 1968.

[19] A.R. Smith. Planar 2-pass texture mapping and warping. *Computer Graphics*, 21(4):263–272, 1987.

[20] G.C. Stockman, S. Kopstein, and S. Benett. Matching images to models for registration and object detection via clustering. *IEEE Trans. Pattern Anal. Mach. Intell.*, 4:229–241, 1982.

[21] G. Wolberg. *Digital Image Warping.* IEEE Computer Society Press, 1990.

[22] G. Wolberg and T.E. Boult. Separable image warping with spatial lookup tables. *Computer Graphics*, 23(3):369–378, 1989.