COMPUTER VISION, GRAPHICS, AND IMAGE PROCESSING **35**, 220–233 (1986)

Algorithms for Subpixel Registration

QI TIAN AND MICHAEL N. HUHNS*

Center for Machine Intelligence, Department of Electrical and Computer Engineering, University of South Carolina, Columbia, South Carolina 29208

Received April 2, 1985; accepted March 27, 1986

This paper presents an analysis of four algorithms which are able to register images with subpixel accuracy; these are correlation interpolation, intensity interpolation, differential method, and phase correlation. The subpixel registration problem is described in detail and the resampling process for subpixel registration is analyzed theoretically. It is shown that the main factors affecting registration accuracy are the interpolation function, sampling frequency, number of bits per pixel, and frequency content of the image. An iterative version of the intensity interpolation algorithm, which achieves maximum computational efficiency, is also presented. Analyses, computer simulations, and experiments for measuring displacements of objects using their speckle images have shown that this algorithm is faster than a direct intensity interpolation algorithm by a factor of more than ten thousand. Using bilinear interpolation and representing pixels by 8-bit samples, a 0.01 to 0.05 pixel registration accuracy can be achieved. © 1986 Academic Press, Inc.

1. INTRODUCTION

Many image processing applications require the registration of pairs of images. For some of these it is acceptable to achieve a registration result with an error of ± 1 pixel; but for others, such as change detection, passive navigation, feature location measurements in remote sensing, image sequence analysis, and nondestructive evaluation, registration results with an error less than one pixel distance are essential [1–6]. For example, in nondestructive evaluation, several images of an object during deformation are typically taken and analyzed. The translations of points located on a closed contour on the surface of the object are measured as accurately as possible. The results yield direct estimates of strain and, if they are sufficiently accurate, can be used to predict structural failures [3, 4]. A similar technique has been used to calculate fluid velocity distributions [5]. In remote sensing [1], a one pixel distance for a Landsat image corresponds to about 80 m distance on the Earth, so that pixel-level registration provides ± 40 m resolution. If an accuracy of 0.1 pixel can be achieved, then ± 4 m resolution can be obtained.

These applications have led to the development of many different algorithms for subpixel registration [6–8]. To date, however, little effort has been devoted to formally defining the subpixel registration problem and systematically comparing previously developed algorithms. This paper addresses these two topics and presents an efficient iterative intensity interpolation algorithm.

In Section 2, four algorithms for subpixel registration are reviewed. Section 3 contains a mathematical analysis of the resampling which is used for intensity interpolation. It is shown that this resampling can be performed by convolving a given discrete reference image with an interpolation function sampled at a higher frequency; the error which results can then be evaluated by comparing the frequency

*Address correspondence to: Michael N. Huhns, 9430 Research Blvd., Austin, TX 78759.

0734-189X/86 $3.00

response of the sampled interpolation function with the frequency response of an ideal low-pass filter.

Section 4 describes an iterative intensity interpolation algorithm for subpixel registration. This algorithm speeds up the direct intensity interpolation method more than ten thousand times. In Section 5, the accuracy of the intensity interpolation algorithm is analyzed as a function of three main factors: sampling frequency, number of bits per pixel, and interpolation function. Then, all of the algorithms for subpixel registration are compared in terms of accuracy, computational cost, and usage. Finally, results for computer simulations and experimental displacement measurements are presented.

2. SUBPIXEL REGISTRATION

This section describes the signal registration problem and four algorithms for achieving subpixel accuracy: correlation interpolation, intensity interpolation, differential method, and phase correlation.

2.1. Signal Registration

A description of the signal registration problem is presented in one dimension, but the results can be easily extended to two dimensions. It is assumed that the signals to be matched differ only by an unknown translation. (In practice, these signals are two images of the same scene, taken at different times, or from different perspectives, or by using different sensors. Besides translation, there may be rotation, noise, and geometrical distortions. Nevertheless, this description reveals the fundamentals of the problem.) Given a discrete signal $f(n)$ with a duration of N samples, as indicated in Fig. 1, another signal $g(n)$ results from translating $f(n)$ by a distance L. The registration problem is, given signals $f(n)$ and $g(n)$, to calculate a displacement value L' which approximates the distance L within a fraction of a sample spacing (pixel).

As also indicated in Fig. 1, instead of using the entire signal $g(n)$, a segment of it with length m, denoted by $gm(n)$, is typically used. The segment is chosen so that the position of greatest interest is located in the middle of the segment. For rigid translations a more accurate result can be achieved by using all of $g(n)$ to perform the matching procedure. However, there are often intensity distortions and geometrical distortions in real images. Using a segment of $g(n)$ increases the tolerance of the matching procedure to these distortions. Also, the number of computations needed is reduced.

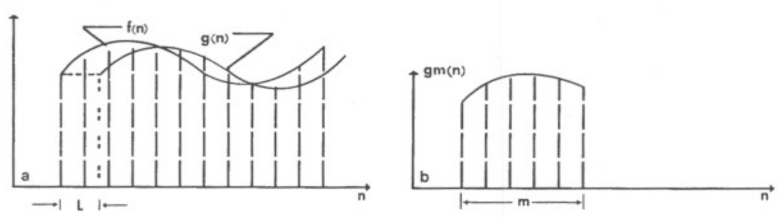

FIG. 1. One-dimensional signals used for subpixel registration: (a) An original signal $f(n)$ and a derived signal $g(n)$ which results from translating $f(n)$ by a distance L. (b) A segment of $g(n)$, $gm(n)$, with a duration of m samples.

The procedure for pixel-level registration is to first calculate the correlation between $f(n)$ and $gm(n)$ as

$$r(k) = \sum_{n=1}^{M} R[f(n+k-1), gm(n)] \tag{1}$$

where $1 \leq k \leq N - M + 1$ and R is a similarity measure, such as a normalized cross-correlation or an absolute difference function. For the $(N - M + 1)$ samples of $r(k)$, there is a maximum value $r(L')$, where L' is the displacement in sampling intervals of $g(n)$ with respect to $f(n)$. The actual displacement L, however, is not necessarily an integer number of sampling intervals.

2.2. Correlation Interpolation

One way to achieve subpixel registration accuracy is to calculate the discrete correlation function between two images, fit an interpolation surface to samples of this function, and then accurately search for the maximum of this surface [2, 9]. When the images are sampled at a high enough frequency, the corresponding discrete correlation function is quite smooth and a second-order interpolation function can provide an accurate representation. A quadratic estimator for the maximum of this second-order function is used in [2] for the analysis of a scan-line jitter that mechanical FLIR's exhibit; it reduces this jitter from 0.5 pixel to less than 0.1 pixel (worst case).

The quadratic estimator can be expressed as

$$X = \frac{P_a - P_b}{2*(2P_m - P_b - P_a)} \tag{2}$$

where P_m is the maximum value of the sampled correlation function, P_b and P_a are the samples to the left and right of P_m, and X is the estimated location of the peak in terms of the sample interval, referenced to P_m. The accuracy of this estimator depends on how well a correlation function around the peak approximates a parabola [2].

2.3. Intensity Interpolation

Subpixel registration can be achieved by using intensity interpolation to create a much denser grid for selected parts of the reference image. A search using the target image is then conducted over these parts. Given a target image of size $M \times M$ and a reference image of size $N \times N$, if a registration accuracy of 0.1 pixel is desired, then the reference image should be interpolated to create a new version with dimensions $(10 \times N) \times (10 \times N)$. Then, a search potentially could cover all $(10N - M + 1)^2$ positions. An iterative algorithm which decreases this computational cost is described in Section 4.

2.4. Differential Method

A differential method is used for estimating 2-dimensional translations to process a sequence of TV images in [6]. The key idea is to relate the difference between two successive frames to the spatial intensity gradient of the first image. Given two images, $f_1(x, y)$ and $f_2(x, y)$, assume that translations of an object centered at (x, y) of image 1 with respect to image 2 are D_x and D_y in the X and Y directions, respectively; then

$$f_2(x, y) = f_1(x - D_x, y - D_y). \tag{3}$$

The frame difference can be expressed as

$$\begin{aligned} I(x, y) &= f_1(x, y) - f_2(x, y) \\ &= f_1(x, y) - f_1(x - D_x, y - D_y) \\ &= \frac{\partial f_1(x, y)}{\partial x} D_x + \frac{\partial f_1(x, y)}{\partial y} D_y \end{aligned} \tag{4}$$

where $\partial f_1/\partial x$ and $\partial f_1/\partial y$ are the partial gradients of $f_1(x, y)$. This relation holds only when translations are very small compared to the spatial change of image gray-levels.

To calculate the translation for one object in the image sequence, a set of intensity difference equations for a small neighborhood of the object, of size $M \times M$ pixels, must be calculated. M usually has a value between 4 and 64, depending on the object involved. There are M^2 simultaneous equations

$$[I] = [G]\,[D] \tag{5}$$

where $[I]$ is the frame difference column matrix, $[G]$ the gradient matrix, and $[D]$ the column vector of translations. Because $[G]$ is not square in general, the pseudoinverse technique is used to solve this equation. Assuming $[G]$ $[D]$ is nonsingular, the translation vector $[D]$ is

$$[D] = \left[[G]^{\mathrm{T}}[G]\right]^{-1}[G]^{\mathrm{T}}[I] \tag{6}$$

where $[[G]^{\mathrm{T}}[G]]$ is 2×2, $[G]^{\mathrm{T}}$ is $2 \times M$, and $[I]$ is $M \times 2$.

The differential method in [6] has been used to estimate motion, which is, in turn, used for motion-compensated temporal filtering to restore and enhance image sequences. This algorithm is faster than the interpolation algorithms, and can be used when an image contains several objects moving with different directions and speeds, but only when the displacements of the objects are small with respect to their sizes.

2.5. *Phase Correlation*

The phase correlation technique has been used to implement a video-rate image correlation processor [7, 8]. It achieves subpixel accuracy with relative insensitivity to scene content, illumination differences, and narrow-band noise.

It is based on the fact that most of the information about the relative displacements of objects between two images is contained in the phase of their cross-power spectrum. If F_1 and F_2 are the discrete 2-dimensional Fourier transforms of sampled images f_1 and f_2, then the correlation C between f_1 and f_2 can be expressed as

$$C = f_1 \otimes f_2 = \mathscr{F}^{-1}\{F_1 F_2^*\} \tag{7}$$

where $\otimes$ denotes convolution, * denotes the complex conjugate, and $\mathscr{F}^{-1}$ denotes the inverse Fourier transform. $F_1 F_2^*$ is normalized to remove dependencies on image content, while leaving dependencies on image shifts. The phase matrix Φ and phase correlation matrix D are then

$$\Phi = \frac{F_1 F_2^*}{|F_1 F_2|} \tag{8}$$

$$D = \mathscr{F}^{-1}\{\Phi\}. \tag{9}$$

In an ideal case of images cyclically translated by L, Φ is a delta function located at L.

To find D with a minimum number of computations, the inverse Fourier transform is first calculated at low resolution. After the peak of this low-resolution version of D is found, a 9×9 point segment of D, centered at the peak, is calculated by using an inverse Fourier transform at full resolution. The peak of this segment is found and a quadratic interpolation of this point with its four surrounding points is used to obtain a final peak position with subpixel accuracy. It was reported that the algorithm provides results with a 0.08 pixel accuracy [7]. The advantage of the method is that it can be used when images are seriously distorted, in either geometry or intensity.

3. INTENSITY INTERPOLATION AND RESAMPLING

As stated above, the intensity interpolation algorithm requires that new versions of the reference image with denser grids be calculated. This process is called resampling and involves two procedures: restoring the original continuous image from a given discrete image; and resampling the continuous image at a higher rate than the one used previously. Registration accuracy depends on how accurately the new reference image approximates the original image [10].

In one dimension, to create a more densely sampled function $f'(n)$ from a given function $f(n)$, it is assumed first that the original continuous function $f(t)$ was band-limited, and that the original sampling frequency was higher than the Nyquist frequency. $f(t)$ can then be restored by convolving a sinc function with $f(n)$, as

$$\begin{aligned} f(t) &= f(m) \otimes \operatorname{sinc}(t) \\ &= \sum_{m=1}^{N} [f(m) * \operatorname{sinc}(t - m * t_0)] \end{aligned} \tag{10}$$

where t_0 is the sample spacing. The resampled function is then

$$f'(n) = f(t) * \sum_{j=1}^{K(N-1)+1} [\delta(n - j * t_1)] \tag{11}$$

where delta represents a unit impulse, $t_1 = t_0/K$ is the new sample spacing, and K is chosen to yield a $(1/K)$th pixel accuracy.

Because the sinc function has infinite length, all points of $f(n)$ would have to be used to calculate $f(t)$. This is time-consuming. Therefore, a finite interpolation function, such as a nearest neighbor, linear, or cubic B-spline, is usually used. Beause the interpolation functions have frequency responses other than the ideal low-pass filter response of the sinc function, aliasing in the restored $f(t)$ occurs. If the interpolation function is denoted by $s(t)$, then

$$\begin{aligned} f'(n) &= \left\{ \sum_{m=1}^{N} [f(m) * s(t - m * t_0)] \right\} * \sum_{j=1}^{N(K-1)+1} [\delta(n - j * t_1)] \\ &= \{f(m) \otimes s(t)\} * \sum_{j=1}^{K(N-1)-1} [\delta(n - j * t_1)] \\ &= f(m) \otimes s(n) \end{aligned} \tag{12}$$

where $s(n)$ represents a sampled interpolation function with sample spacing t_1.

Achieving a precise registration depends on how accurately $f'(n)$ approximates $f(t)$. This comparison can be done in the frequency domain. The Fourier transform of (12) is

$$F'(nw_0) = F(nw_0) * [S(w) \otimes \Sigma\{\delta(n * w_1)\}]. \tag{13}$$

Analyzing how the interpolation function and resampling frequency affect the accuracy is the same as analyzing the frequency distribution of $s(n)$. Because the frequency response of a sampled interpolation function can be obtained by simply repeating the response of the interpolation function at every $n * w_1$ frequency position, and if the sampling rate for the interpolation function is high enough so that aliasing can be ignored, then the main effect of the sampled interpolation function on the frequency response of $f'(n)$ is attenuation inside the passband.

As indicated in Fig. 2, among the three interpolation functions, the cubic B-spline has the best passband response. A nearest neighbor interpolation should be ex-

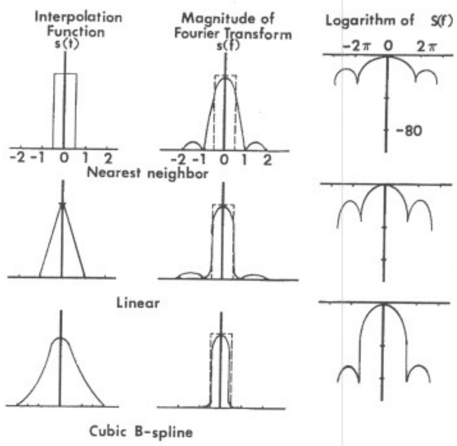

FIG. 2. Several interpolation functions and the magnitudes of their Fourier transforms.

cluded from consideration because it causes a maximum 0.5 pixel shift of the interpolated signal, and this is not acceptable for subpixel accuracy. Experiments with computer-generated sinc-function-like images indicate that linear interpolation provides about 0.005 pixel accuracy. But, in experiments with real images using bilinear interpolation, system noise reduces the accuracy to 0.05 pixel. Also, the computation time is lengthy. Therefore, it is better to use a simple interpolation method, unless the accuracy is mainly limited by this method and the increased computations which result are acceptable.

4. ITERATIVE INTENSITY INTERPOLATION

As described in the last section, reference images with a denser grid are created by a resampling procedure. However, a complete search of a more densely sampled image can be very time-consuming. For example, to achieve a 0.05 pixel accuracy for a 512 × 512 image requires the search of a 10240 × 10240 image. However, this can be greatly reduced by conducting the search in an iterative manner. During each iteration, only a small part of the reference image is created at an increased resolution; a best match location at this resolution level is then found by searching. During the next iteration, a smaller part of the reference image, located around the best match location, is calculated at an even higher resolution and searched for the best match. This continues until a final accuracy is reached.

4.1. Notation

Before describing the algorithm, several parameters need to be defined. The initial resolution, R_0, is the number of pixels that the target image is shifted between each search point of the reference image. This can be chosen as one pixel or several pixels, depending on the images to be registered. It should be chosen, however, so that a result near the correct match location, with a distance smaller than R_0 can be located. The final resolution, R_f, determines the desired registration accuracy. The resolution ratio, K, specifies the improvement in accuracy achieved by each iteration, represented as $K = R_{i-1}/R_i$. The number of iterations, Num, needed to reach the final resolution can be calculated from

$$R_0 = R_f * (K^{\mathrm{Num}}). \tag{14}$$

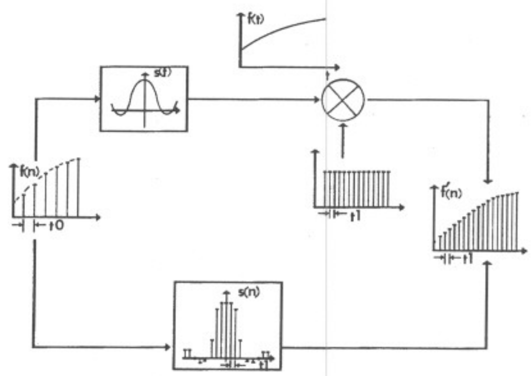

FIG. 3. Resampling can be performed by convolving $f(n)$ with a sampled interpolation function $s(n)$.

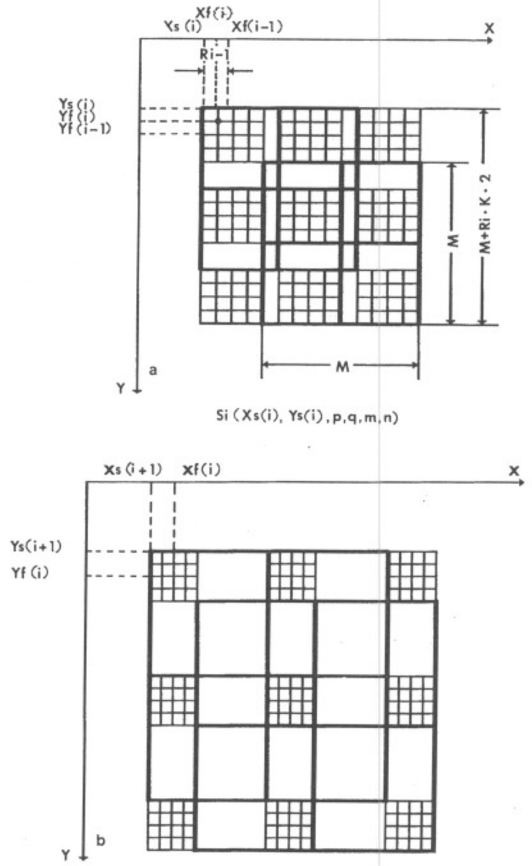

FIG. 4. Resampled reference images for the ith and $(i + 1)$th iterations in a new coordinate system (X, Y), where the search begins at $(X_s(i), Y_s(i))$ for the ith iteration, and the match position is $(X_f(i), Y_f(i))$. ($M = 3$ and $K = 2$ are used for this figure.) (a) Iteration i. (b) Iteration $i + 1$.

Figures 4a and 4b show different versions of the reference image for the ith and $(i + 1)$th iterations. These versions have the same number of pixels, $[(2K + 1) * M]^2$, but they have different side lengths. The side length for the ith iteration is $M + 2R_i * K$. A target image of dimension $M \times M$ is shown overlaid (in boldface) at both the first (top left) and last (bottom right) search positions. The figures indicate all pixel positions where interpolated values have to be calculated.

The target image is denoted by $T(m, n)$ and is the same for all iterations. The reference image for the ith iteration is denoted by $S_i(X_s(i), Y_s(i), p, q, m, n)$, where m and n are integer indices of an $M \times M$ subimage of the reference image, and where p and q are integer indices for the top-left corner of the subimage. The real values $X_s(i)$ and $Y_s(i)$ are the top left corner of this image. A subimage of S_i with the same dimensions $M \times M$ as the target image T is represented by $S_i'(X(i), Y(i), m, n)$, where m and n are indices of the subimage itself, $(X(i), Y(i))$ are its top-left corner coordinates, and the distance between neighboring pixels is the same as the target image—one standard pixel length. The coordinates of the

mth row and nth column of $S_i'(X(i), Y(i), m, n)$ in the X–Y coordinate system are then

$$(X, Y) = [X(i) + m - 1, Y(i) + n - 1]. \tag{15}$$

Thus, for a subimage where the top-left corner is located at (p, q), the intensity of the mth row and nth column pixel of the subimage is calculated from

$$\begin{aligned} S_i'(X_s(i) + (p-1)*R_i, Y_s(i) + (q-1)*R_i, m, n) \\ = \text{interpolated}\{S(X_s(i) + (p-1)*R_i + m - 1, \\ Y_s(i) + (q-1)*R_i + n - 1)\} \end{aligned} \tag{16}$$

where $m = 1, 2, \ldots, M$; $n = 1, 2, \ldots, M$; $p = 1, 2, \ldots, (2K+1)$; and $q = 1, 2, \ldots, (2K+1)$.

The coordinates for S in most cases are not integer values, but its intensity can be interpolated by using an algorithm such as the bilinear interpolation shown below. For simplicity, $s(0,0)$, $s(0,1)$, $s(1,0)$, and $s(1,1)$ represent the intensities of the four nearest pixels to (X, Y). Then

$$\begin{aligned} X &= X_s(i) + (p-1)*R_i + m - 1 \\ Y &= Y_s(i) + (q-1)*R_i + n - 1 \\ s(0,0) &= S(\text{INT}[X], \text{INT}[Y]) \\ s(0,1) &= S(\text{INT}[X], \text{INT}[Y] + 1) \\ s(1,0) &= S(\text{INT}[X] + 1, \text{INT}[Y]) \\ s(1,1) &= S(\text{INT}[X] + 1, \text{INT}[Y] + 1) \\ b &= s(0,1) - s(0,0) \\ c &= s(1,0) - s(0,0) \\ d &= s(1,1) - s(0,0) - s(1,0) - s(0,1) \end{aligned}$$

and

$$\begin{aligned} S_i'(X_s(i) + (p-1)*R_i, Y_s(i) + (q-1)*R_i, m, n) \\ = s(0,0) + b*(X - \text{INT}[X]) + c*(Y - \text{INT}[Y]) \\ + d*(X - \text{INT}[X])*(Y - \text{INT}[Y]). \end{aligned} \tag{17}$$

(Note. The INT operator takes the integer part of its operand.)

4.2. *Iterative Procedure*

Suppose that at iteration $(i-1)$ a position $(X_f(i-1), Y_f(i-1))$ is determined as the match location at the search resolution R_{i-1}; then during the next iteration the search should be performed around this position with a neighborhood of size $[2(R_{i-1})] \times [2(R_{i-1})]$. Based on the relation $R_{i-1} = K * R_i$, there are $(2K+1)^2$ positions to be searched. In Fig. 4, a 5×5 search area is indicated, where $M = 3$ and $K = 2$.

After a match location in the $(i-1)$th iteration is determined, the top-left position of the reference image for the ith iteration is determined. To avoid missing a match location in the ith iteration, the search is conducted over a square area, where the $(i-1)$th match location is located at its center and both of its sides have length $2(R_i - 1)$. Thus, the top-left position of the new reference image is

$$X_s(i) = X_f(i-1) - K * R_i = X_f(i-1) - R_{i-1} \tag{18}$$

$$Y_s(i) = Y_f(i-1) - K * R_i = Y_f(i-1) - R_{i-1}. \tag{19}$$

The new reference image has the same dimensions as before, but it is denser since the distance between neighboring pixels is R_i, where $R_i = R_{i-1}/K$. From the above expressions, in the ith iteration the cross-correlation function is

$$\begin{aligned}\mathrm{CORR}(p,q) &= \sum_{m=1}^{M}\sum_{n=1}^{M}\big|T(m,n) - S_i'[X_s(i) + (p-1)*R_i, Y_s(i) \\ &\qquad + (q-1)*R_i, m, n]\big|\end{aligned} \tag{20}$$

where p and $q = 1, 2, \ldots, (2K+1)$.

After comparing $(2K+1)^2$ cross-correlation coefficients, a location with a minimum $\mathrm{CORR}(p,q)$ is determined and denoted by $p_{\min}$ and $q_{\min}$. A match location for the ith iteration is

$$X_f(i) = X_s(i) + (p_{\min} - 1) * R_i \tag{21}$$

$$Y_f(i) = Y_s(i) + (q_{\min} - 1) * R_i. \tag{22}$$

The process is repeated until a desired R_f is reached. Note that although the target image and all versions of the reference image have the same dimensions, M^2 and $((2K+1)*M)^2$, respectively, they are at a different resolution. The former has a resolution of one pixel, while the latter has a resolution which increases with each iteration.

4.3. Number of Computations

The total number of target and reference images to be compared to achieve a final resolution of R_f can be shown to be

$$T = \frac{(2K+1)^2 * \log[(N-M)/2 * R_f]}{\log(K)}. \tag{23}$$

For fixed values of N, M, and R_f, this is a minimum when $K = 2$. This means that it is most efficient when the search area for each iteration is a $(2K+1)*(2K+1) = 5 \times 5$ neighborhood.

5. ACCURACY ANALYSIS AND COMPARISONS

Simulations using computer-generated images and displacement measurements using real speckle images have been conducted to evaluate the accuracy of the intensity interpolation algorithm, and to compare it with the other algorithms. In these simulations, to avoid noise and distortion problems, 1-dimensional signals and 2-dimensional images were computer generated. In this way, the translation between two given signals was exactly predetermined. Test results could then be compared with this known translation to determine a method's inaccuracies. The experiments with real images were conducted as follows: a board was painted flat white to generate a speckle pattern when illuminated by a laser. The board was then translated a known distance by an X–Y translation table.

5.1. *Accuracy of Intensity Interpolation Algorithm*

Theoretically, if (1) the sampling frequency is high enough, (2) a sinc interpolation function is used to restore the signal, and (3) the signal is digitized with enough bits to accurately represent it, then the intensity interpolation algorithm yields perfect registration results. However, errors occur in practice because these three assumptions are violated.

A sampling frequency below the Nyquist frequency causes aliasing errors. For interpolation this means that because the samples are too far apart in comparison with changes in the image signal, interpolating to calculate intensities between these samples results in errors. Thus, a presmoothing of the sampled signals is often helpful in improving the accuracy, especially when the image to be sampled is very detailed. A 1-dimensional sawtooth waveform has been used to demonstrate this effect. Two-point-average and three-point-average low-pass filters were used to remove high frequencies, which occur near the sharp points on the waveforms. Table 1 indicates that (1) the accuracy can be improved by one order and (2) the performance of the three-point-average filter is much better than the two-point-average filter. In this simulation, real-valued signals were used to test the effects of filtering independent of the effects of finite-length words.

Second, the finite number of bits per pixel adds to the registration error. To test the magnitude of this error, two straight lines with slope = 5 are separated by 0.611111 pixel and then sampled. If both signals are represented by real-valued samples, then the error is 0.000005 pixel; if they are represented by 8-bit integers, then the error is about 0.01 pixel. For a pair of sine waveforms, errors for using real and integer-valued signals are 0.0005 and 0.003, respectively. This means that to achieve registration accuracies better than 0.01 pixel more than 8 bits is required to represent each sample.

In measuring displacements of a real object, we digitized an image, then moved the object a known distance in the X direction and digitized a second image. The displacements measured by using iterative intensity interpolation are shown in

TABLE 1
Test Results to Evaluate Prefiltering

	Displacement = 0.6111	Test error
no filtering	0.6186	0.0075
2-point filter	0.6149	0.0038
3-point filter	0.6106	0.0005

TABLE 2
Measurement Results Using Speckle Images

	Values (in pixels)					
Actual	0.310	3.100	4.650	6.200	7.750	9.300
Measured	0.326	3.025	4.607	6.145	7.761	9.338
Error	−0.016	0.075	0.043	0.055	−0.011	−0.038

Table 2. The known displacement in the X direction for each step is 0.001 in. and for the Y direction is 0.000 in.

Another experiment was performed to show the system's overall accuracy limits. A VICOM image processing system hosted by a VAX-11/780 was used. For two images without any translation the measured translation in the vertical direction was about 0.03 pixel and in the horizontal direction was about 0.01 pixel. These values are the overall errors caused by the equipment and the algorithm. Thus, computer simulations and experiments measuring the displacements of real objects indicate that by using bilinear interpolation, 8 bits per pixel, and a sampling rate greater than the Nyquist frequency, a 0.01 to 0.05 pixel accuracy can be expected.

5.2. Intensity Interpolation vs Other Algorithms

In Section 2, the correlation interpolation, differential, and phase-correlation algorithms are described. Here they are compared in terms of accuracy, speed, and ease of use on computer-generated images. These images are

$$f_1(i, j) = 120 \frac{\mathrm{SIN}[K_x(I - 50.1)]}{K_x(I - 50.1)} * \frac{\mathrm{SIN}[K_y(J - 50.1)]}{K_y(J - 50.1)} \tag{24}$$

$$f_2(i, j) = f_1(i + D_i, j + D_j) \tag{25}$$

where $D_i = 0.11111$, $D_j = 1.11111$, $K_x = 0.4$, and $K_y = 0.2$. Different values for K_x and K_y cause the images to have different shapes in X and Y. 16×16 images are used for all tests.

The correlation interpolation method seems easiest to implement. In one dimension, a peak of the correlation function can be calculated using second-order interpolation. In two dimensions, there are several methods which can be used to find an approximate location of a peak. A 5-point, 6-point, or 9-point neighborhood near the peak can be used to fit a 2-dimensional equation which approximates the correlation surface; then, from this equation, a peak can be located. Alternatively, separate 1-dimensional interpolations in the X and Y directions can be used to determine the coordinates of the peak. This method can be used if the correlation peak is symmetric. In performance, correlation interpolation is less accurate than intensity interpolation while having a much lower computational cost. Table 3 contains the experimental results.

The differential method is appropriate for applications where objects have only small translations, because it is based on an assumption that during translation the intensity gradients of an image do not change. Computational costs depend on the

TABLE 3

Results of Different Algorithms Applied to Sinc Images

	Intensity interpolation	Differential method	Correlation interpolation	Phase correlation
X_{err}	0.0029	0.013	0.045	0.092
Y_{err}	0.006	0.010	0.052	0.127

dimensions of the area used for the calculation, typically from 4×4 to 64×64 pixels. Compared with other methods, its computational cost is the lowest, but its use is limited to the small displacement case. Generally, its accuracy is lower than intensity interpolation, but higher than correlation interpolation.

The phase-correlation method is most easily used when an entire image frame is uniformly shifted. Because image content is normalized out, the algorithm is not sensitive to either geometric distortions or noise. Experiments show that even when two images have less than 70% common area, the algorithm still provides satisfactory results. The accuracy, however, is the lowest among the subpixel registration algorithms.

6. CONCLUSIONS

Subpixel registration can be used in a variety of application areas because it provides a way to accurately measure displacements of individual points of a plane, without any contact and disturbance. In this paper, several algorithms for subpixel registration have been both theoretically and experimentally analyzed, and compared in terms of accuracy, computational cost, and usage. In decreasing order of registration accuracy, the algorithms are intensity interpolation, differential method, correlation interpolation, and phase correlation. Intensity interpolation is, however, the most time consuming.

An iterative intensity interpolation algorithm has been described which reduces this computation time. It is most efficiently performed when optimum parameters for the start search resolution, final search resolution, and resolution ratio are chosen. Several criteria for choosing these parameters have been discussed. It also has been shown that the algorithm can be implemented most efficiently when, for each iteration, the search is conducted over a 5×5 grid. In [4], a hill-climbing algorithm is incorporated with the iterative intensity interpolation method, and a factor of 10 improvement in speed is achieved without affecting the accuracy.

An analysis of the resampling process needed for intensity interpolation was presented, because it determines whether a denser version of the reference image can be generated accurately. This resampling process can be performed by convolving the original discrete reference image with an interpolation function sampled at a higher frequency. The best interpolation functions have frequency responses as close as possible to that of an ideal low-pass filter. A computer simulation has shown that, in the ideal case where computer-generated images have no distortion or noise, bilinear intensity interpolation yields less than a 0.005 pixel error.

Experiments for measuring the displacements of objects by using their speckle images demonstrate that when bilinear interpolation is used in resampling a reference image and image pixels are represented by 8-bit samples, an accuracy of 0.01 to 0.05 pixel can be achieved for the intensity interpolation algorithm. This error is mainly caused by system noise which arises in the electronics of the image digitizer. The registration accuracy can be further improved by reducing noise in the imaging system, increasing the number of bits per pixel, or presmoothing the images to remove some of the high-frequency details.

REFERENCES

1. N. A. Bryant (ed.), *Proceedings of the NASA Workshop on Registration and Rectification*, Leesburg, Va., June 1982.
2. V. N. Dvorchenko, Bounds on (deterministic) correlation functions with applications to registration, *IEEE Trans. Pattern Anal. Mach. Intell.* **PAMI-5**, No. 2, 1983, 206–213.
3. W. H. Peters, W. F. Ranson, M. A. Sutton, T. C. Chu, and J. Anderson, Application of digital correlation methods to rigid body mechanics, *Opt. Eng.*, **22**, 1983, 738–742.
4. Q. Tian, M. N. Huhns, A Fast Iterative Hill-Climbing Algorithm for Subpixel Registration, in *Proceedings, 7th Int. Conf. on Pattern Recognition*, Montreal, Canada, July 30 to August 2, 1984, Vol. 1, 13–16.
5. Z.-H. He, M. A. Sutton, W. F. Ranson, and W. H. Peters, Two-dimensional Fluid Velocity Measurements by Use of Digital Speckle Correlation Techniques, *Experimental Mechanics*, **24**, 1984, 117–121.
6. T. S. Huang, *Image Sequence Analysis*, pp. 303, Springer-Verlag, Berlin/Heidelberg, 1981.
7. J. J. Pearson, D. C. Hines, Jr., and S. Golosman, Video-rate image correlation processor, in *Applications of Digital Image Processing*, Proc. SPIE Vol. 119, pp. 197–205, 1977.
8. C. D. Kuglin and D. C. Hines, The phase correlation image alignment method, *Proceedings, IEEE International Conference on Cybernetics and Society*, September 1975, 163–165.
9. P. E. Anuta, Spatial registration of multispectral and multitemporal digital imagery using fast Fourier transform techniques, *IEEE Trans. Geosci. Electron.* **GE-8**, 1970, 353–368.
10. J. A. Parker, R. V. Kenyon, and D. E. Droxel, Comparison of interpolating methods for image resampling, *IEEE Trans. Med. Imaging* **MI-2**, No. 1, 1983, 31–39.