NOTICE

CERTAIN DATA CONTAINED IN THIS DOCUMENT MAY BE DIFFICULT TO READ IN MICROFICHE PRODUCTS.

CONF-9011125--5

DE91 004777

Subpixel measurement of image features based on paraboloid surface fit

Shaun S. Gleason, Martin A. Hunt, W. Bruce Jatko

Oak Ridge National Laboratory*
P.O. Box 2008, Oak Ridge, Tennessee, 37831

ABSTRACT

A digital image processing inspection system is under development at Oak Ridge National Laboratory that will locate image features on printed material and measure distances between them to accuracies of 0.001 in. An algorithm has been developed for this system that can locate unique image features to subpixel accuracies. It is based on a least-squares fit of a paraboloid function to the surface generated by correlating a reference image feature against a test image search area. Normalizing the correlation surface makes the algorithm robust in the presence of illumination variations and local flaws. Subpixel accuracies better than 1/16 of a pixel have been achieved using a variety of different reference image features.

1. INTRODUCTION

An algorithm has been developed at Oak Ridge National Laboratory that can be used to locate a variety of features or objects in a digitized image to subpixel accuracies. The algorithm uses a set of normalized correlation data[1] generated by correlating a stored reference template against a test image search area that is believed to contain the feature of interest. A three-dimensional paraboloid surface is fitted to the data to interpolate the precise point of maximum correlation or degree of similarity.

Applications for this type of searching algorithm are numerous. Almost every image-processing/machine vision system is concerned with the spatial registration of the inspected object, and this algorithm is robust enough to find a large variety of objects. This algorithm also has the advantage that high accuracies can be achieved without large magnification optics at the front end of the system because of its subpixel measurement capabilities.

This paper presents the theoretical background for the algorithm as well as a step-by-step discussion of how to actually implement it. Also, experimental results from extensive testing done with a machine vision system will be presented.

2. IMAGE CORRELATION

Consider a reference template of image data, $R(x,y)$, and a test image search window, $T(i,j)$, as shown in Fig.1. One way to locate the feature of interest in $T(i,j)$ is to load the template $R(x,y)$ with the image data that represents the feature of interest. Then place $R(x,y)$ at every possible location on $T(i,j)$ and calculate the correlation, or degree of similarity, between the two. Depending on the type of correlation used, a maximum or a minimum correlation value will occur when the template lies directly on top of the matching feature of interest.

The type of correlation can be as simple as a pixel-by-pixel subtraction of the reference and test templates. An ideal match in this case would be zero. The type of data used in the proposed algorithm is generated by calculating the normalized correlation between the templates. The square of the normalized correlation, c, between $R(x,y)$ and $T(i,j)$ for some position (l,m) in T where $l \le i-x$ and $m \le j-y$ is given by[1]

*Operated by Martin Marietta Energy Systems, Inc. for the U.S. Dept. of Energy under contract No. DE-AC05-84OR21400

$$c^2(l,m) = \frac{\left[\sum_x \sum_y R(x,y)\, T(l+x, m+y)\right]^2}{\sum_x \sum_y \left[R(x,y)\right]^2 \sum_x \sum_y \left[T(l+x, m+y)\right]^2}. \quad (1)$$

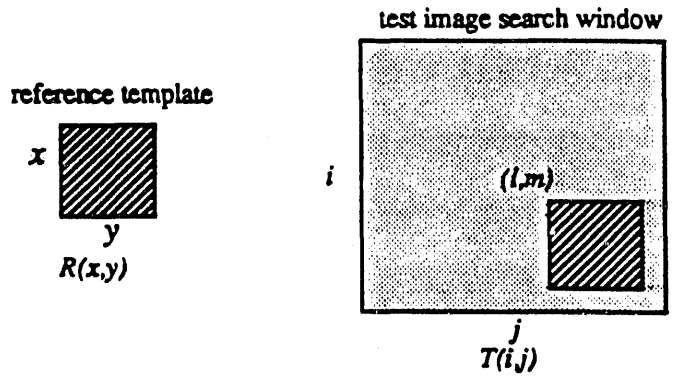

Fig. 1. Correlation concept illustration.

In this application of equation (1), the substitution $z(l, m) = c^2(l, m)$ was made to provide a larger falloff in the correlation value. The normalized correlation equation (1) will produce a value ranging from zero to 1 with one meaning a perfect match and zero meaning no correlation at all.

To find the best match between the two features to the nearest pixel, the maximum normalized correlation value for the whole window must be found. Evaluating equation (1) at *(i-x)(j-y)* different points can be very expensive in terms of computation time. Several different types of correlation could be used to quickly find the neighborhood of the best registration.[2] Once the general location of the best match is known, the normalized correlation can be used to pinpoint the feature of interest to the nearest pixel. The subpixel registration algorithm presented here is based on the normalized correlation values obtained at the peak of the surface as well as the eight surrounding points as shown in Fig. 2. The following description of the subpixel algorithm assumes that the nearest integer pixel match has already been located.

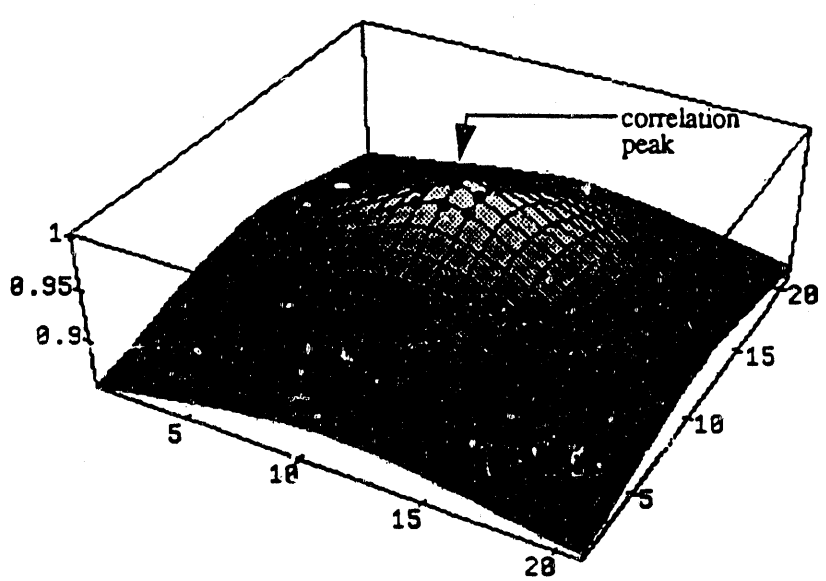

Fig. 2. Correlation surface with peak and surrounding points marked.

3. SUBPIXEL IMAGE CORRELATION

This section of the paper develops the subpixel technique for the general case. The application of the technique to three different cases of image features will then be discussed. The first and simplest case considers an object that is symmetric with respect to the x and y axes and lies on a zero background. The second case expands the first case to an object that is not symmetric. The final case describes the effects of having a nonzero background in the scene. The assumption is made in developing the algorithm that the template containing the feature of interest is square to simplify the math, but this procedure can be generalized to include any size rectangle.

3.1 General case reference and search areas

Consider a square template, R, of size n, that is given by

$$R = \begin{bmatrix} r_{0,0} & r_{0,1} & \cdots & r_{0,n-1} \\ r_{1,0} & r_{1,1} & & \cdots \\ \cdots & & \cdots & \cdots \\ r_{n-1,0} & \cdots & \cdots & r_{n-1,n-1} \end{bmatrix}. \tag{2}$$

Now the search for R will be made in a square search window, T, of size m where m is larger than n and is given by

$$T = \begin{bmatrix} t_{0,0} & \cdots & \cdots & \cdots & t_{0,m-1} \\ \cdots & & \cdots & \cdots & \\ \cdots & t_{i-1,j-1} & \cdots & t_{i-1,j+n} & \cdots \\ \cdots & \cdots & R & \cdots & \cdots \\ \cdots & t_{i+n,j-1} & \cdots & t_{i+n,j+n} & \cdots \\ & \cdots & \cdots & \cdots & \\ t_{m-1,0} & \cdots & \cdots & \cdots & t_{m-1,m-1} \end{bmatrix}. \tag{3}$$

It is important to notice at this point that the object R lies somewhere within the region of T as shown in equation (3). If R is placed directly on top of its match in T, the correlation would have a value of 1 since the image features are identical by definition. Then, to generate the other eight surrounding points, R is translated one pixel in all eight directions (left, right, up, down, and diagonally) from its current position and the normalized correlation is found according to equation (1). For example, if R were slid up one pixel from its original position, the normalized correlation, $z_{1,0}$, would be given as

$$z_{1,0} = \frac{\left(\sum_{\alpha=0}^{n-1} (r_{0,\alpha})(t_{i-1,j+\alpha}) + \sum_{\alpha=0}^{n-1} (r_{1,\alpha})(r_{0,\alpha}) + \ldots + \sum_{\alpha=0}^{n-1} (r_{n-1,\alpha})(r_{n-2,\alpha})\right)^2}{S_R \left(S_{T^2} + \sum_{\alpha=0}^{n-1} (t_{i-1,j+\alpha})^2 - \sum_{\alpha=0}^{n-1} (r_{n-1,\alpha})^2\right)}, \tag{4}$$

where the subscripts of z represent the x and y direction of template movement and

$$S_{R^2} = \sum_{x=0}^{n-1} \sum_{y=0}^{n-1} r_{xy}^2 \tag{5}$$

$$S_{T^2} = \sum_{x=i}^{i+n-1} \sum_{y=j}^{j+n-1} t_{xy}^2. \tag{6}$$

The numerator of equation (4) is simply the dot product between R and the region of T that lies directly underneath it. The denominator consists of the dot product of R against itself multiplied by the dot product of the test region covered by R against itself. The second term in the denominator of equation (4) is found by taking the test region dot product $S_T{}^2$ and adding the new row of T covered as a result of sliding the template up one pixel, and then subtracting out the row at the bottom of the template that was uncovered (see Fig. 3). It is also helpful to note that $S_R{}^2 = S_T{}^2$ in this case because T contains the identical feature, R.

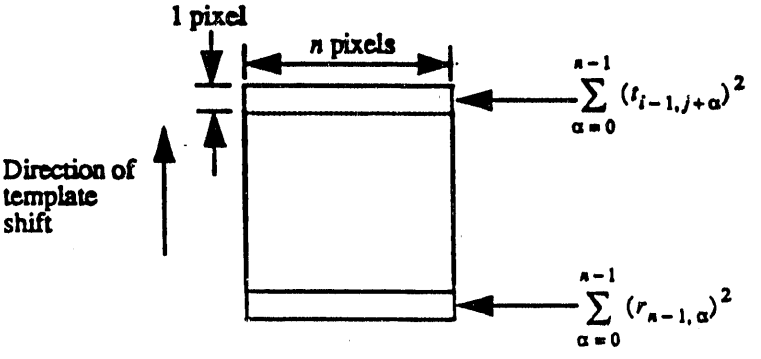

Fig. 3. Calculating the $z_{1,0}$ correlation term.

It is evident that if R was moved down one pixel from its original position that the resulting correlation value, $z_{-1,0}$, would be

$$z_{-1,0} = \frac{\left(\sum_{\alpha=0}^{n-1} (r_{0,\alpha})(t_{1,\alpha}) + \sum_{\alpha=0}^{n-1} (r_{1,\alpha})(t_{2,\alpha}) + \ldots + \sum_{\alpha=0}^{n-1} (r_{n-1,\alpha})(t_{i+n,j+\alpha})\right)^2}{S_{R^2}\left(S_{T^2} + \sum_{\alpha=0}^{n-1} (t_{i+n,j+\alpha})^2 - \sum_{\alpha=0}^{n-1} (r_{0,\alpha})^2\right)}. \tag{7}$$

The general case equations (4) and (7) are used to illustrate the theory behind the subpixel surface fit.

3.2 Symmetric reference feature on a zero background

Consider the case where the reference template given in equation (4) is symmetric with respect to the x and y axes. That is,

$$r_{x,i} = r_{n-1-x,i} \qquad \forall x = 0, 1, \ldots, n-1, \tag{8}$$

and

$$r_{i,y} = r_{i,n-1-y} \qquad \forall y = 0, 1, \ldots, n-1. \tag{9}$$

In other words, the left half of the reference is a mirror image of the right, and the top is a mirror image of the bottom. Examples of this type of image feature would be a square, circle, or rectangle of uniform intensity.

Also in this case, assume that the background of T is all zero. It can then be shown that $z_{1,0}$ is exactly equal to $z_{-1,0}$ by looking at equations (4) and (7). Notice that the terms in the numerator of each equation that contain any t entries will go to zero because of the zero background. This makes the numerators of equation (4) and (7) equal. Also notice that the terms in the denominator involving t fall out. Finally, the terms in each denominator that subtract a row of the reference template are equal because of the symmetry of the reference template. The top row, $r_{0,\alpha}$, is the same as the last row, $r_{n-1,\alpha}$, as illustrated in equation (8). This leaves

$$z_{1,0} = z_{-1,0} = \frac{\left(\sum_{\alpha=0}^{n-1} (r_{0,\alpha})(r_{1,\alpha}) + \sum_{\alpha=0}^{n-1} (r_{1,\alpha})(r_{2,\alpha}) + \ldots + \sum_{\alpha=0}^{n-1} (r_{n-2,\alpha})(r_{n-1,\alpha})\right)^2}{S_R^2\left(S_R^2 - \sum_{\alpha=0}^{n-1} r_{0,\alpha}^2\right)}. \tag{10}$$

Note that r_0 in the denominator can be replaced by r_{n-1}. Because of the x and y axis symmetry, it can be shown that in general

$$z_{i,j} = z_{-i-j}, \tag{11}$$

which says that all of the normalized correlation pairs on opposite sides of the peak must be equal to each other for the case of a symmetric feature on a zero background considered here.

3.3 Nonsymmetric reference on zero background

This section shows that the type of reference template can be expanded to include features that are not symmetric, and the paraboloid surface fit will still be very accurate under certain conditions.

Consider the case where T has a zero background, but the reference template is any general feature. The only difference between this case and the previous one is that equations (8) and (9) no longer hold true. In this case, $z_{1,0}$ and $z_{-1,0}$ are not equal because the subtracted terms in the denominators of (4) and (7) are not the same. That is,

$$\sum_{x=0}^{n-1} (r_{0,x})^2 \neq \sum_{x=0}^{n-1} (r_{n-1,x})^2. \tag{12}$$

Notice that even though these terms are no longer equal, their value, which is the dot product of one row (n pixel intensities) in the template, would in most cases be very small compared to the S_T^2 term, which consists of the dot products of the remaining n-1 rows summed together. So if the reference template is sufficiently large, $z_{1,0}$ will be approximately equal to $z_{-1,0}$. Most of the testing done with the algorithm has been with n greater than or equal to 20 pixels. The analysis just described can be applied to all of the eight points surrounding the peak. The difference between $z_{-1,-1}$ and $z_{1,1}$ will vary slightly more because moving R in a diagonal direction introduces a new row as well as a new column to be subtracted from terms in the denominators of the equations. But in the final analysis, the extra row and column of data have a negligible effect on the total value of the correlation. An example of a correlation data set is given in Fig. 4.

3.4 Including nonzero background effects

This section uses the same approach as the last section to show that nonzero background in T will have a small effect on the symmetry of the correlation surface if the reference template is large enough.

As the nonsymmetric template R is moved over a template T with a nonzero background, the denominator effects introduced in the last section still apply, but new terms are introduced in the numerator of the correlation equation because the reference template moves onto some nonzero background as it is translated

one pixel in each direction. The terms involving the background that were previously dropped in equations (4) and (7) must now be included.

The contributions of these extra rows and columns to the final result are minimal compared to the dot product of the remaining n-1 rows and columns. As R moves further away from the best match (the correlation peak), more and more rows and columns are introduced to add to the difference in slope on opposite sides of the correlation peak. But since this algorithm requires only that the reference be moved one pixel in each direction around the peak, it is accurate to model opposite points as being equal.

Fig. 4 shows an excerpt from a normalized correlation data file that shows the peak of the surface (having a value of 1) and several other values around the maximum. These data were taken by correlating a nonsymmetric reference feature against a test search area with nonzero background. These data show that the approximations given in sections 3.3 and 3.4 hold true for this case. The correlation values on opposite sides of the peak vary only in the fourth significant digit even with the nonsymmetric feature and nonzero background.

```
0.951022 0.959333 0.964479 0.967207 0.964564 0.958956 0.950671
0.959285 0.969520 0.976865 0.980583 0.976653 0.969197 0.959110
0.965085 0.977052 0.987385 0.993811 0.987366 0.976713 0.964791
0.966576 0.979440 0.991520 1.000000 0.991275 0.978717 0.966824
0.964989 0.977848 0.987364 0.993366 0.986755 0.975718 0.963891
0.959598 0.969570 0.976486 0.980873 0.976468 0.968355 0.958349
0.951502 0.959323 0.964331 0.967243 0.965018 0.959177 0.950923
```

Fig. 4. Example of normalized correlation data.

4. MODELING A SURFACE TO FIT THE DATA

4.1 Constructing the paraboloid

The model of the surface that fits the set of normalized correlation data points is now considered. In a one-dimensional case it can be shown that if there exists some maximum point, $z_{0,0}$, and points on either side of $z_{0,0}$ that are of equal value and are equidistant from $z_{0,0}$, then the smallest degree curve that will pass through the three points and have a mathematical maximum at $z_{0,0}$ is a parabola of second degree[3] (see Fig. 5).

It follows that one-dimensional parabolas can be fitted through the maximum correlation value of the surface in Fig. 2 and any two other points on opposite sides of the peak. The only difference between all of the parabolas is their widths due to different surface gradients depending on the correlation pair ($z_{i,j}$, $z_{-i,-j}$) being considered.

This behavior can be modeled in three dimensions by a paraboloid of one sheet that has the following general equation.

$$z(x, y) = ax^2 + by^2 + cxy + dx + ey + f. \tag{13}$$

This surface can be visualized by taking a one-dimensional parabola and rotating it about its axis of symmetry while allowing the width to vary such that it forms a three-dimensional paraboloid with an elliptic base as shown in Fig. 5.

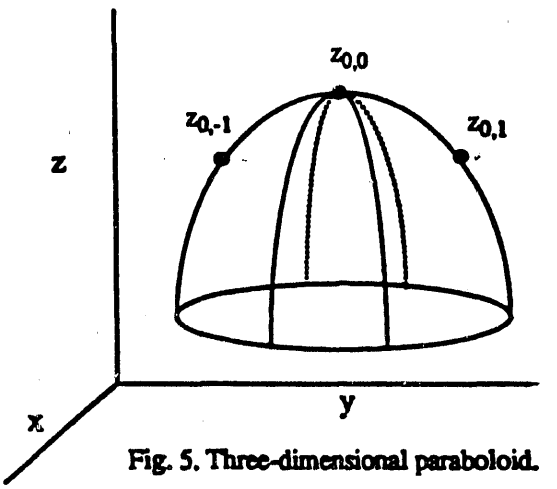

Fig. 5. Three-dimensional paraboloid.

4.2 Fitting the paraboloid to the correlation peak

The solution for the coefficients of the paraboloid can be found by solving the system of equations

$$Ax = b, \tag{14}$$

where

$$A = \begin{bmatrix} x_0^2 & y_0^2 & x_0y_0 & x_0 & y_0 & 1 \\ x_1^2 & y_1^2 & x_1y_1 & x_1 & y_1 & 1 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ x_8^2 & y_8^2 & x_8y_8 & x_8 & y_8 & 1 \end{bmatrix}, \tag{15}$$

$$x = \begin{bmatrix} a \\ b \\ c \\ d \\ e \\ f \end{bmatrix}, \tag{16}$$

and

$$b = \begin{bmatrix} z_0 \\ z_1 \\ \cdots \\ z_8 \end{bmatrix}. \tag{17}$$

The correlation data points in b have been redefined from $z_{0,0}$, $z_{0,1}$, $z_{1,0}$, etc., to z_0 through z_8 to be consistent with the A matrix format. Because there are only six unknown values in the coefficient matrix, x, and nine correlation data points are supplied by the correlation maximum and the eight surrounding neighbors as

given in b, the system of equations is overdetermined. Because A is a rectangular matrix and cannot be inverted in the normal fashion to solve for x, a least-squares regression must be used. This approach has an advantage in that all nine points can contribute to the position and shape of the resulting paraboloid.

Application of a pseudo-inverse method[4] can be applied to the system. The first step is to multiply both sides of equation (14) by A^T, the transpose of A.

$$A^T A x = A^T b. \tag{18}$$

Gaussian elimination could be applied to equation (18), or since $A^T A$ is a square matrix, it can be inverted by a numerical algorithm to solve for x as shown in equation (19).[5]

$$x = (A^T A)^{-1} A^T b. \tag{19}$$

Once the coefficients of the surface are known, the maximum of the paraboloid can be found by taking the gradient of $z(x,y)$ in equation (13) with respect to x and y and setting them equal to zero as follows:

$$\frac{\delta z}{\delta x} = 2ax + cy + d = 0; \tag{20}$$

$$\frac{\delta z}{\delta y} = 2by + cx + e = 0. \tag{21}$$

The values for x and y at the peak can be solved for directly as

$$x = \frac{(2ab - ce)}{(c^2 - 4ab)}, \tag{22}$$

and

$$y = \frac{(2ae - dc)}{(c^2 - 4ab)}. \tag{23}$$

Note that this is a closed form solution for the maximum. No iterative steps are involved, and the execution time is therefore deterministic.

The application of the algorithm can be summarized as follows:

1. Determine the best integer pixel match between the reference template and the feature of interest in the search window.
2. Find the normalized correlation at the peak and eight surrounding points (z_0, ..., z_8) using equation (1).
3. Set up the matrices for the system as shown in equations (14) through (17) using the correlation data points and their associated (x, y) positions.
4. Solve for the coefficients of the paraboloid, a, b, ..., f using a numerical method on equation (18).
5. Use the coefficients and equations (22) and (23) to calculate the position of the surface maximum.
6. Add this subpixel value to the integer pixel location found in step 1 to precisely locate the feature.

5. EXPERIMENTAL RESULTS

The primary application of the algorithm described in the previous sections is to locate a unique feature within an image on a printed sheet. A mean accuracy of at least 1/8 of a pixel (pixel size is 0.0058") is

required for our particular application. Several simulations of the algorithm indicated that an accuracy of better than 1/16 of a pixel was possible. An experiment was designed to determine the accuracy and repeatability of the algorithm using actual imagery.

A test stand was constructed for use in the experiment that consisted of two translation stages mounted on an optical table. The translation stages were assembled to allow travel in the x and y directions and were positioned with precision micrometers. A mechanical fixture to hold the printed sheet was mounted on top of the translation stages. The images were digitized with a high-resolution (1320 x 1035) charge coupled device (CCD) camera.

The first step in the experimental procedure was to acquire a reference image for use in the correlation process. After storing this image in memory, a unique feature was selected as the reference template for correlation purposes. The position of the reference feature was defined as the origin of a cartesian coordinate system. The translation stages were used to shift the graphic material in 0.001" increments in x, y, and combinations of both. A total of 560 images were acquired at 10 different locations. At each new position the subpixel cross correlation algorithm was used to estimate the amount of shift from the starting position. The calculated shift was compared to the actual shift as measured by the micrometer. This experiment was designed to determine both the accuracy and the standard error by precisely positioning the printed sheet and by acquiring many images at each location.

The results of this experiment are shown in Fig. 6. In this plot, the axes represent the fraction of a pixel which the printed sheet was shifted in the x and y directions. The circles are centered at the ten positions where data were acquired, and their radius represents 1/16 of a pixel. The squares are centered at the mean position calculated using the subpixel correlation, and their width and height correspond to one standard error in the x and y directions, respectively.

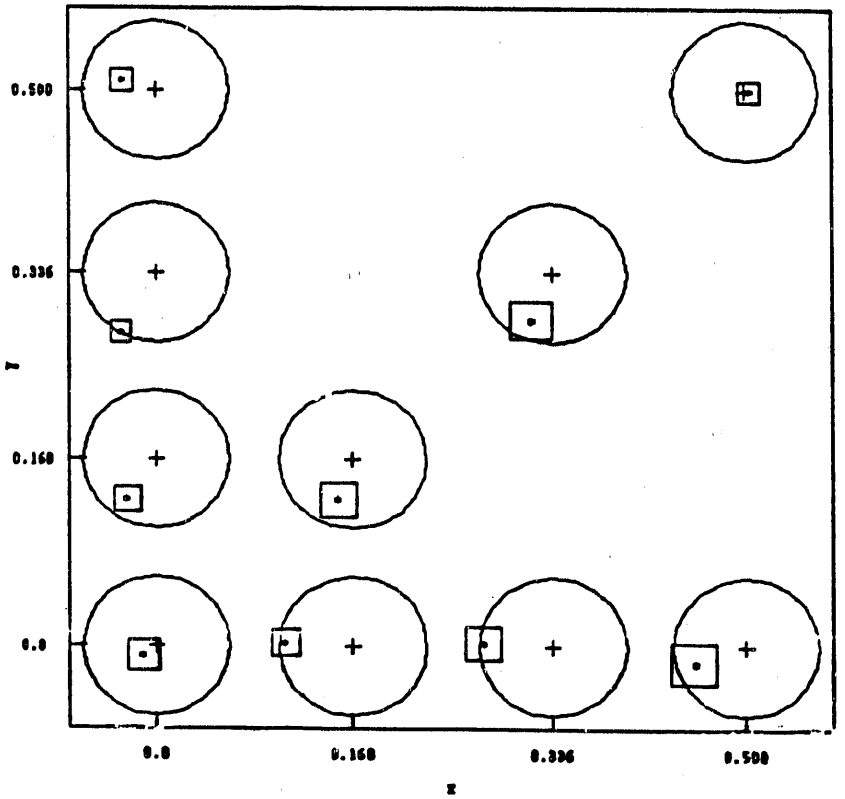

Fig. 6. Plot of calculated translations of a printed graphic image, circles centered at ideal location with a radius of one-sixteenth pixel and squares centered at mean of calculated location with width and height representing one standard deviation in x and y, respectively.

As can be seen from Fig. 6, the mean calculated location was within 1/16 of a pixel at each actual position. Another important conclusion is that the magnitude of the standard error was on average less than 1/32 of a pixel. Results from this experiment proved that the algorithm was sufficient for the needs of the application.

6. CONCLUSION

A subpixel registration algorithm has been developed for a digital inspection system at Oak Ridge National Laboratory. The inspection system will be used to make measurements of printed materials in order to precisely locate particular image features. The algorithm presented has been shown to be accurate to 1/16 pixel on a variety of different image features, so it can be applied to many different types of machine vision inspection systems where spatial registration is critical. The algorithm can be programmed in closed-form which allows a deterministic execution time and can be fast enough to implement in an on-line inspection environment without a large investment in expensive image-processing hardware.

7. REFERENCES

1. R. C. Gonzalez and P. Wintz, *Digital Image Processing*, pp. 425-426, Addison-Wesley Publishing Company, Inc., Reading, Mass., 1987.

2. D. I. Barnea, "A Class of Algorithms for Fast Digital Image Registration," *IEEE Trans. Comput.*, vol. C-21, pp. 179-186, 1972.

3. P. Lancaster, K. Salkauskas, *Curve and Surface Fitting*, Academic Press, Inc., San Diego, CA, 1986.

4. G. W. Stewart, *Introduction to Matrix Computations*, Academic Press, Inc., San Diego, CA, 1973.

5. W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes in C*, pp. 37-45, Cambridge University Press, New York, 1988.

DISCLAIMER

END

DATE FILMED

02/27/91