Compensating for Centroid Errors Due to Surface Tilt and Lens Distortion

William Hoff, Lance Gatrell, Dan Layne, Guy Bruno, Cheryl Sklair
Martin Marietta Astronautics Group
P.O. Box 179
Mail Stop 4372
Denver, CO 80201

Accurately locating features in images is essential to many problems in computer vision, including object recognition, pose estimation, and camera calibration. Image centroid features have been commonly used due to their perceived advantages of accuracy, ease of processing, and robustness. However, non-linear effects in the projection of object features to the image plane can cause significant errors in the centroid (i.e., the projected centroid of the object feature does not coincide with the centroid of the projected image region), which in turn can cause significant errors in pose estimation and camera calibration. Two of these effects are the tilt of the object's surface away from the image plane, and the presence of radial lens distortion. In this work, we analyze the sources and magnitudes of centroid errors, and describe methods for compensating for the errors. Three methods were developed to predict centroid errors: a simulator, an approach based on numerical integration, and a neural network. The first two methods provide accurate results but are too slow for real time applications. The neural network, on the other hand, provides less accurate results but is much faster than the first two and is suitable for real time applications.

1. INTRODUCTION

This paper describes a novel technique for significantly improving the accuracy of centroid-based image features. Accurately locating features in images is essential to many problems in computer vision, including object recognition [Besl85], pose estimation [Hara89], and camera calibration [Tsai88]. The accuracy of extracted image feature locations directly influences the accuracy of derived object pose estimates, the accuracy of derived internal camera parameters, and the reliability of model-based object recognition. In many applications, it is necessary to determine feature locations to subpixel accuracy to meet given requirements for accuracy of derived results.

One type of feature that can be found to subpixel precision is the centroid of a two-dimensional image region. Such an image region can arise from the projection of a three-dimension object feature onto the image plane. These object features can be visually distinctive markings that are naturally occurring or artificial. A high contrast planar shape such as a dark circle on a light background is an example of a visually distinctive marking. The image projection of a feature such as this can be segmented using simple and fast image processing techniques such as thresholding and connected-component labeling.

In many applications such as manufacturing and inspection, which deal with man-made objects, the objects already possess visually distinctive markings, or else one has the freedom to place markings on them. These markings, which are also called *landmarks* or *fiducials*, are positioned in precise known locations relative to each other and to the object. Tasks such as object recognition and registration are then simplified since only the fiducials need be extracted from the image. However, the fiducials must be located accurately since all derived results are based on their image locations.

Although any shape can be used, circular features have been claimed by a number of researchers to have very desirable properties, such as low spatial quantization error, invariance to translation and roll, and insensitivity to image noise [Path89, Bose90, O'Gor91]. The centroid of a circular feature can typically be found to a precision of about 0.02 to 0.06 pixels, depending on

its size, image noise, *etc* [Skla91]. Circle centroid features have been used by many researchers for object recognition, pose estimation, and camera calibration [Tsai89, Lenz89, Tsai88, Corn87, Skla90, Hoff91, Davi87, Abid90]. In our own work, we have used circular features in an application for NASA's Flight Telerobotic Servicer project. This project required very accurate visual pose estimation (better than 0.1 inch and 0.3°) in order to verify the positional accuracy of the robot arm while it was being tested in orbit.

There are many factors which contribute to error in the estimated location of a feature [Skla91]. Some of these are random errors such as image gray scale noise and spatial quantization noise, which cannot be compensated for. Other errors are systematic, such as errors in camera calibration parameters, that can be reduced through the use of an adequate camera model and calibration procedure [Tsai87, Tsai88].

We have analyzed another source of systematic error that has not been reported in the past, and also a method for compensating for this error. This error arises from nonlinear effects in the projective transformation from object points to image points. As a result, the centroid of the projected image region does not in general coincide with the projection of the centroid of the planar marking.

One cause of nonlinearity is the perspective projection, which is nonlinear when the surface of the planar marking is tilted so that it is not parallel to the image plane. The error due to surface tilt can be several pixels in magnitude, depending on the size and pose of the planar marking.

For example, Figure 1 shows a perspective projection of a planar circular region, tilted approximately 35° away from the image plane. This is an actual digitized image from an RS-170 CCD video camera with an 8 mm lens. One can see that the image region corresponding to the circle is not a true ellipse (as would be the case for a linear or affine transformation), but is asymmetrically elongated in one direction. At the center of the circle is a small white dot. The centroid of the projected image region is marked by the crosshair at the lower left. For this particular case, the centroid of the projected image region (marked by the left crosshair) differs by about 14 pixels from the projected circle center (marked by the right crosshair over the small white

dot). Although this example uses an artificially large image region for illustrative purposes, which exaggerates the centroid error, even small image regions can have centroid errors that are larger than that which would be due to image or spatial quantization noise alone.

The fact that the centroid error occurs when the surface is tilted is especially troublesome for three-dimensional recognition, pose estimation, and calibration applications that make use of co-planar target features. First of all, in these applications one cannot in general constrain the target surface to be parallel to the image plane. Secondly, some algorithms in fact require that the target surface not be exactly parallel to the image plane, since they experience singularities in these cases [Tsai87, Abid90]. Finally, some pose estimation algorithms are actually most accurate when the target surface is tilted at an angle of 45° from the image plane [Kris91, Yuan89].

Another cause of nonlinearity, that is independent of surface tilt, is lens distortion. With lens distortion, even if the surface of a circular feature is parallel to the image plane, it will still not project to a circular region in the image. For example, radial lens distortion will cause the region to be asymmetrically elongated along the radial direction. Radial lens distortion is present in most off-the-shelf camera lenses, but can be calibrated [Tsai87]. Again, depending on the size and location of the image region, the centroid error due to lens distortion can be significantly larger than that which would be due to image or spatial quantization noise alone.

The centroid error due to these effects, surface tilt and lens distortion, is entirely deterministic and can be predicted, given knowledge of the location and tilt of the surface marking, its shape and size, and the camera model. This suggests an approach in which object pose (and the camera model, if desired) is first estimated without regard for the centroid errors. These approximate results are used to predict the centroid errors. Then the centroids are adjusted to compensate for the errors due to surface tilt and lens distortion, and a more refined and accurate pose (and camera model, if desired) is recomputed. Note that one must have an *a priori* model of the object and the target features in order to use this technique.

We have implemented just such a technique in our work and have used it to improve centroid accuracy and object pose estimates. In fact, we applied it to the example shown in Figure

1 to compensate for the centroid error and calculated a more accurate estimate of the circle center. This improved estimate is marked by the crosshair on the right, which as can be seen in Figure 1, lies directly over the true circle center, which is marked by the small white dot. In this example, we first calculated the pose of the circle, using the monocular projection of the four small outer black circles, and the known geometric model of the four circles [Hung85]. Using this pose and the known model of the large circle, we predicted the centroid error of the large circle and corrected the original extracted image centroid by this amount.

The rest of this paper describes the methods we have developed to predict centroid error, given an estimate of the surface tilt and the camera model, and gives example results. We have developed two independent methods — one based on simulation and the other based on numerical integration. The results from these two methods agree to a high degree of precision. We have also developed a third method based on a neural network, that was trained using the numerical integrator. The neural network gives less accurate results, but is computationally much faster than the first two methods, and is suitable for real time pose estimation and object tracking.

All three methods predict centroid errors for circular features of any size and at any position and orientation. Although we have exclusively used circle features in our work, our method is applicable to planar features of any shape, as long as the shape is known. However, the neural network would have to be retrained for features other than circles.

Thus, our method is fast, applicable to any planar features, and demonstratably improves pose estimation. To our knowledge, there has been no other previous work on this specific subject. Although other researchers have studied centroid errors due to quantization and noise, this is the first work that we know of that has analyzed centroid errors due to surface tilt and lens distortion. Our results should be useful to anyone who needs to perform accurate three-dimensional vision metrology based on the image projection of planar features.

2. NOTATION AND STATEMENT OF PROBLEM

This section formally states the problem of centroid error prediction, and introduces notation that will be used consistently through the next three sections. The next three sections will describe the simulator, the numerical integrator, and the neural network, respectively.

Let R be a two dimensional planar closed region of arbitrary shape and size, embedded in a three dimensional space (Figure 2). Define a three dimensional orthonormal coordinate frame $\{R\}$ attached to this region, with coordinates (u, v, w). Let the origin of this frame, O_R, be located at the center of mass (centroid) of the region. Also restrict the axes (u,v) to lie in the plane of R, which means that the third axis, w, is perpendicular to the plane.

Now consider an imaging sensor that is projecting the region onto an image plane. We will use a pinhole camera model for the purpose of discussion, although there is nothing that precludes more complex camera models. Define a three dimensional orthonormal coordinate frame $\{C\}$ attached to the camera, with coordinates (X,Y,Z). Also define a two dimensional coordinate system $\{I\}$ on the image plane, with coordinates (x,y). If the camera is properly calibrated, there is a known mapping g from three dimensional points (X,Y,Z) to the image plane (x,y); *i.e.*, g: $\Re^3 \rightarrow \Re^2$. For example, in the case of the pinhole camera, the mapping is simply the perspective projection equations

$$g(X,Y,Z) = (x,y) = \left(f\frac{X}{Z}, f\frac{Y}{Z} \right) \tag{1}$$

where f = focal length. Here, we have assumed that the origin of $\{C\}$, O_C, is at the center of projection of the pinhole camera, and the origin of $\{I\}$ is at the optical center; that is, the intersection of the image plane with the perpendicular to O_C.

Let the transformation from the region's coordinate frame $\{R\}$ to the camera's coordinate frame $\{C\}$ be denoted as H. Specifically, if P_R is a point whose coordinates are in the $\{R\}$ frame, H transforms it so that its coordinates are in the $\{C\}$ frame. If using homogeneous coordinates, $P_R = (X_R, Y_R, Z_R, 1)^T$, $P_C = (X_C, Y_C, Z_C, 1)^T$, and

$$P_C = \mathrm{H}\, P_R \tag{2}$$

where H is a 4x4 homogeneous transformation matrix. Given these definitions and relationships, the projection of a single point in region R onto the image plane is given by

$$(x,y) = g\ (\mathrm{H}\ P_R) \tag{3}$$

Let the image projection of the origin of $\{R\}$, which is the projection of the point O_R, be denoted as ${}^I O_R$. The projection of the entire region R onto the image plane is a region ${}^I R$ such that

$${}^I R = \left\{(x,y) : (x,y) = g(\mathrm{H}P_R),\ \forall P_R \in R\right\} \tag{4}$$

Assume that the image on the image plane is segmented so that the projected region ${}^I R$ is extracted exactly. (We will ignore all real world error sources such as spatial quantization, gray level noise, optical blurring, *etc*; since our goal is to isolate the effect of surface tilt and camera model.) Let the centroid of ${}^I R$ be denoted as ${}^I C_R$. This point will not in general be equal to the point ${}^I O_R$.

We can now state the problem as follows: Given knowledge of the size and shape of region R, an estimate of its transformation with respect to the camera H, and the mapping function $g(X,Y,Z)$, determine the two dimensional error vector between the image projection of its centroid, ${}^I O_R$, and the centroid of the projected region, ${}^I C_R$.

3. SIMULATOR

We first developed a simulator to predict the centroid errors because it was easy to implement, although the running time is quite slow. The simulator performs a "brute-force" projection of the fiducial feature to the image plane and directly calculates the centroid of the synthesized projected image region. Although we have only worked with circular features, any planar shape can be used.

The simulator takes as input the estimated (X_0, Y_0, Z_0) location of the circle's center in the camera's coordinate frame $\{C\}$, the estimated surface normal direction, the radius ρ of the circle, and either a pinhole camera model or a Tsai camera model. The parameters of the Tsai camera model include the focal length, the inter-pixel spacing of the CCD elements in X and Y, the size of

the digitized image in pixels, radial lens distortion, and the location of the optical center in the image. In the discussion for the remainder of this section, we use a pinhole camera model.

The simulator computes the first order moments (centroid) of the circle's projection using the standard equations (assuming a binary image):

$$\bar{x} = \left(\frac{1}{A}\right)\left(\sum_{(x,y)\in {}^{I}R} x(\Delta A)\right), \quad \bar{y} = \left(\frac{1}{A}\right)\left(\sum_{(x,y)\in {}^{I}R} y(\Delta A)\right) \tag{5}$$

where A = area of the projected image region ${}^{I}R$, and ΔA = the area of the discrete elements used in the summation.

Given a point $P_I = (x,y)$ on the image plane, the vector from P_I through the origin O_C (focal point) is $\mathbf{v} = (-x, -y, -f)$, where f = focal length. A point $P_R = (X, Y, Z)$ is on the vector $\mathbf{v}$ if there exists some scalar t such that $t\,\mathbf{v} = P_R$. The point P_R is also on the plane of the fiducial region if it satisfies the equation $AX + BY + CZ = D$ (where (A, B, C) is the unit surface normal of the plane, and D is the perpendicular distance from the plane to the origin). Substituting $t\,\mathbf{v} = P_R$ into the equation of the plane, we have

$$t = -D/(Ax + By + Cf) \tag{6}$$

and thus P_R can be computed. A point P_R is inside the circle fiducial if it is within ρ of the center. A synthetic projection of the circle is generated by this method, and then the centroid of the synthesized image region is found.

Spatial quantization error can be reduced by increasing the resolution of the pixel grid [Path89]. In our implementation, each pixel is subdivided to a user specified resolution, such as a 14x14 or a 254x254 grid. Each subdivided point is used in the computation of the centroid moments. Figure 3 shows the effect of increasing the resolution. Specifically, it shows how the centroid changes as the subpixel resolution is increased, for a typical range of configurations[1].

---

[1]The circle had a radius of 1 cm, the pan and tilt angles were combinations of 0°, 15°, 30°, and 45°, the focal length was 1 cm, and the CCD pixels were square and 0.0013 cm on a side. The circle center location was randomly varied in X and Y, with a fixed Z distance of 10 cm, for a total of 896 cases altogether.

The numbers displayed are the maximum differences between the computed centroid at the finest 254x254 subpixel resolution and the computed centroid at coarser resolutions (only the x difference is shown, the y difference is essentially the same). The results indicate that increasing the resolution beyond about 10x10 has very little effect on the centroid. For subpixel resolutions of greater than 10x10, the centroid results are within 0.001 pixel. Beyond 30x30, the results are within 0.0005 pixel.

Although the simulator was easy to implement and is very general, it is too slow to use for a real time (*i.e.*, video rate) application. The running time to predict the centroid error for a particular case depends on the size of the circle and also on the subpixel resolution chosen. As an example, for the configuration that was used to generate the results in Figure 3, and at a subpixel resolution of 34x34, the running time on a Solbourne 5/501 workstation was about 14.9 seconds. This workstation is compatible with a Sun SPARC station, and is rated at about 22 MIPS and 3.4 MFlops.

In an effort to devise a faster centroid error predictor, and also to verify the results of the simulator, we developed an approach that was based on numerical integration. This is discussed in the next section.

4. NUMERICAL INTEGRATOR

This approach was based on evaluating the following integrals:

$$\bar{x} = \left(\frac{1}{A}\right)\iint_{I_R} x\,dxdy,\ \bar{y} = \left(\frac{1}{A}\right)\iint_{I_R} y\,dxdy,\ A = \iint_{I_R} dxdy \tag{7}$$

where ${}^{I}R$ is the image region of the projected fiducial (in our case, a circle). Since the shape and size of the projected image region is unknown, we change the problem so that we can integrate over the known fiducial region, R, instead. This can be done using the well-known change of variable theorem for transforming multiple integrals [Buck78]:

Theorem: Let T be a continuously differentiable transformation from 2-space into 2-space, with $T(u,v) = (x,y)$, which is 1-to-1 in an open set Ω with a non-zero Jacobian ($J(p) \neq 0$) throughout Ω. Let D^* be a closed bounded set in xy-space which is the image under T of a set $D \subset \Omega$. Let f be a continuous function on D^*. Then,

$$\iint_{D^*} f(x,y)dxdy = \iint_{D} f(T(u,v))|J(u,v)|dudv \tag{8}$$

To cast our problem in the framework of this theorem we first replace function f with the continuous moment functions $f(x,y) = x^m y^n$, where $m,n \in \{0,1\}$. The transformation T is the composition of the camera perspective transformation g with the homogeneous transform H describing the pose of the target in camera coordinates (as given in Equation 3 of Section 2). In this work, we used only the pinhole camera model for the transformation g. We define the domain, D, of the transformation T, to be the set of points (u,v) inside the boundary of the target. The co-domain of T, $T(D) = D^*$, is therefore the subset of the image plane corresponding to the projection of the target. The determinant of the Jacobian of T was evaluated directly using the Mathematica® package to be:

$$|J(u,v)| = \begin{vmatrix} \frac{\partial x}{\partial u} & \frac{\partial x}{\partial u} \\ \frac{\partial x}{\partial u} & \frac{\partial x}{\partial u} \end{vmatrix} = \frac{f^2(\hat{z}_R \cdot r_0)}{(z_C)^3} \tag{9}$$

where $\hat{z}_R$ is the unit vector describing the z-axis of the target's coordinate frame (the target surface normal), r_0 is the vector to the center of the target O_R (in camera coordinates), z_c is the z-component of the point (u,v) in camera coordinates, and f is the camera focal length.

In order to apply the change of variable theorem we must show that all its conditions on T are satisfied. First, observe that the transformation T is continuously differentiable when $z_c(u,v) \neq 0$ for every point on the target. This simply means that the target in camera coordinates does not intersect the image plane. To show that T is an injection we notice that no two points on a planar surface project to the same image point unless the surface normal is orthogonal to the line of sight

from the camera's focal point to a surface point. This condition corresponds to degenerate views resulting from positioning the target such that only its edge is visible. This constraint is explicitly obtained from the condition $|J(u,v)| \neq 0$ which occurs only if $\hat{z}_R \cdot r_0 \neq 0$.

We may now state the result applied to the case where the target is a circular region of radius ρ as the following:

$$\iint_{D^*} x^m y^n \, dx\,dy = f^2(\hat{z}_R \cdot r_0) \int_{-\rho}^{\rho} \int_{-\sqrt{\rho^2-v^2}}^{\sqrt{\rho^2-v^2}} \frac{x(u,v)^m y(u,v)^n}{z_C(u,v)^3} \, du\,dv \tag{10}$$

where $x(u,v)$ and $y(u,v)$ are the image-plane coordinates of the projection of the target point (u,v), using the expressions given in Section 2. Integrals of this form can be evaluated using standard numerical quadrature techniques. We implemented a solution using an IMSL® routine[2], and found that the results agreed very closely with those from the simulator.

Figure 4 shows a plot of the difference between the centroid error computed by the simulator and the numerical integrator. The data is from the 896 cases plotted in Figure 3, but this time showing the maximum difference between the integration result and the simulator result at the various resolutions. The same pinhole camera model was used for both the simulator and the numerical integrator. Again, only the x difference is shown; the y difference is essentially the same. The maximum difference between the integration and simulation results, at the simulator sub-pixel resolution of 254x254, is 0.0000185 pixels (2.405E-7 mm). Since the integrator and the simulator agree so closely at the fine resolution, it is not surprising that the plots shown in Figures 3 and 4 are nearly identical.

The numerical integrator is much faster than the simulator and its accuracy does not depend on the resolution. For the same configuration used in Section 3, the running time on the same

---

[2]We used the IMSL routine DTWODQ, with the ERRABS parameter (absolute accuracy desired) set to 1.0E-12, ERRREL (relative accuracy desired) set to 1.0E-12, and IRULE set to 2 (specifying a Gauss-Kronrod quadrature rule with 10-21 points). The ERREST (estimate of the absolute value of the error) output was less than 1.0E-13 for all results.

workstation was about 2.7 seconds (as compared to 14.9 seconds for the simulator). However, this is still not fast enough for real time (*i.e.*, video rate) operation. We thus developed the neural network predictor described in the next section in order to achieve the necessary speed.

4. NEURAL NETWORK

Neural networks can be trained to approximate underlying multivariable functions. The approximation is accomplished using sample input-output data to adjust network parameters during supervised training. Several network architectures for function approximation are described in the literature. Approaches with feed-forward nets include multi-layer backpropagation [Lape87], radial basis function nets using Gaussian activations [Park91], and CMAC (Cerebellar Model Articulation Controller) networks with B-spline receptive fields [Lane91]. Although we evaluated some of these network models, another network model called Multivariate Adaptive Regression Splines (MARS) produced better results [Frie91a and Frie91b] for this application. MARS estimates cubic spline basis functions with recursive partitioning of the input space, uses statistical learning (least squares) instead of gradient descent, and predicts performance on test data during training using a cross-validation model to improve values of network parameters.

Function approximation with spline (local approximating polynomial) methods [DeBo78] has advantages over other methods, such as capturing local information, and splines are most successful when the input space is partitioned appropriately. After partitioning the input space into separate regions, cubic (or other degree) polynomials are constructed in each region such that the approximating function and its first two derivatives are continuous on the boundaries. The difficulty with applying conventional spline methods, especially for multivariate functions, is determining a good partition of the input space. The tensor-product spline approach partitions each input axis with k points ("knots") and the tensor product of the subintervals is formed, resulting in $(k+q+1)^n$ basis functions, where q is the polynomial degree (3 in this paper), and n is number of

input variables. For most applications this is too many basis functions, causing computational and weight-fitting problems.

The MARS algorithm uses statistical training and recursive partitioning to reduce the high number of tensor-product splines to a reasonable number of basis functions. In the adaptive spline net there are m basis functions B_m, each with weight w_m. The output of the entire network is the weighted sum (11) which approximates the desired function $f(x_1, x_2, ..., x_n)$.

$$\hat{f}(\bar{x}) = \sum_{m=0}^{M} w_m B_m(\bar{x}), \quad \bar{x} = (x_1, x_2, \ldots, x_n) \tag{11}$$

$$B_m(\bar{x}) = \prod_{k=1}^{K_m} (x_{j(k)} - t_{kj})_+^q \tag{12}$$

$$(x_j - t_{kj})_+^q = \begin{cases} 0, & x_j \le t_{kj} \\ (x_j - t_{kj})^q, & x_j \ge t_{kj} \end{cases} \tag{13}$$

Each basis function (12) is a product of simple functions, called truncated power functions, given by (13), where t_{kj} is the location of the k^{th} knot on the j^{th} axis, and K_m is the number of factors in the product (basis function interaction level). The net is an ordered set of interconnected units, with each unit receiving all system inputs $(x_1, x_2, ..., x_n)$. Each adaptive spline unit produces two outputs $B_l(t - x_j)_+^q$ and $B_l(x_j - t)_+^q$ that are available for input selection by all succeeding units. A bias unit is also available to all units. A simple schematic of a spline net is shown in Figure 5.

From the large number of possible basis functions, the training selects a small subset through statistical variable subset selection. The training goal is to choose values for network parameters (weights, knot locations, and input variables for each unit) that minimizes future prediction error, which is estimated from the generalized cross-validation model selection criterion

$$GCV = \frac{1}{N} \sum_{i=1}^{N} (y_i - f_i)^2 \Big/ \left[1 - \frac{5N_u + 1}{N}\right]^2 \tag{14}$$

where N is the number of training samples, y_i is desired output, f_i the approximation, and N_u is the number of units. Training is semi-greedy, with units considered in order. The GCV criterion

is minimized with respect to the parameters of the m^{th} unit and all previous weights. This optimization is performed by least squares, repeating the process until a specified maximum number of units have been added. Weight elimination is then applied to select an optimal set of weights. Results with MARS show the adaptive spline approach, with the GCV criterion adjusting net parameters during training based on expected test performance, can produce system models that yield highly accurate function approximations for system identification. In addition, the MARS results also provides insight into how the system works, for example, by showing which input variables contribute most to the solution, and how the variables interact. Understanding results and interpreting models from other networks such as backpropagation is more difficult.

Training and testing data were generated with the IMSL-based numerical integrator described in the previous section. Normally seven input variables are used to predict centroid error: focal length; the estimated (X_0,Y_0,Z_0) location of the circle center; the radius of the circle; and the pitch and yaw angles (surface tilt). (Roll is not needed since the circles are symmetric.) These seven input variables were reduced to five by scaling the centroid error results in units of focal length, and by scaling X_0, Y_0, and radius in units of Z_0. Input variables were constrained to typical ranges for circle sizes and the field of vision of the camera. Ranges of the scaled inputs are shown in Table 1.

Table 1. Ranges of Input Variables

Input Variable	Minimum Value	Maximum Value
X_0 (in units of Z_0)	–0.4	0.4
Y_0 (in units of Z_0)	–0.4	0.4
Radius (in units of Z_0)	0.02	0.1
Pitch (degrees)	–40.0	40.0
Yaw (degrees)	–40.0	40.0

Two outputs are produced: centroid errors in x and y directions. The outputs are given in units of focal length. In the training data, these both ranged from –0.006604 to 0.006604 (units of focal length). The input sampling scheme yields key features in about 100 samples, while remaining samples have slight variations to provide local detail, which is important in this application. Test data was generated using different sampling schemes, to assess if training was biased by sampling. Median centroid error values are 3 magnitudes smaller than the extreme values, causing difficulties for alternative training approaches, such as backpropagation which doesn't account for all local information. Several training and testing files were used, representing the entire workspace as well as selected subspaces. Input and output data were normalized for faster training. Two spline networks were used: one for x-error and one for y-error, each having 5 inputs, 1 ouput, and the number of spline units determined by MARS parameters. These two nets are independent and can operate in parallel, although they can also be combined into 1 network with two outputs.

Training time with MARS on the Solbourne workstation ranged from 15 minutes for small nets to 3 hours for large nets with large training files. MARS training time was a fraction of the time required for backpropagation nets to converge with less accuracy on same files (although second-order and/or orthogonal pre-processing methods can be used to speed up backpropagation convergence). The trained MARS network performed very fast, computing x-error and y-error in less than 10 msec. Accuracy results are shown in Table 2. All results used cubic splines with up to 3 variable interactions allowed in each basis function.

Accuracy on test data is very good, and in some cases actually better than training accuracy, due in part to the effects of GCV on training. Some training bias is observed by comparing tests (B) and (C). Samples for test (B) were generated similar to the training data (same method, different points over entire input space), while data for test (C) was generated in a different manner in a subspace. Training case (H), and test cases (I) and (J) used the same files as cases (A), (B) and (C). By using more spline units, higher accuracy is obtained and training bias is nearly eliminated. Training and testing results for y-error are similar to those shown in Table 2.

Table 2. MARS Training and Testing Results for x-error

Case	Train/ Test	# of Data Points	# of Spline Units	GCV	RMS	Max	% < 0.0001 abs error
A.	tra	1000	23	7.5E-3	7.5E-5	2.9E-4	82.6
B.	tst	500	23	-	7.7E-5	2.7E-4	80.6
C.	tst	576	23	-	1.1E-5	4.4E-4	75.5
D.	tra	3125	26	5.4E-3	7.0E-5	3.2E-4	84.0
E.	tst	576	26	-	5.0E-5	1.7E-4	94.8
F.	tra	4000	23	3.0E-3	5.3E-5	3.2E-4	93.1
G.	tst	576	23	-	4.2E-5	2.1E-4	96.2
H.	tra	1000	65	8.7E-4	1.8E-5	7.2E-5	100
I.	tst	500	65	-	2.5E-5	1.5E-4	99.6
J.	tst	576	65	-	2.7E-5	1.8E-4	98.8

From the table, we conclude that this initial neural network system can predict centroid error to within about $1x10^{-4}$ units of focal length. The network can be refined to produce higher accuracy. For the typical values that were used in Sections 2 and 3, the focal length was 1 cm and the pixels on the image plane were square and measured 0.0013 cm on a side. The error of $1x10^{-4}$ thus corresponds to about 0.077 pixels. The accuracy to which the neural net can predict centroid errors is thus about equal to the precision of which centroids can be extracted from images, meaning that it is accurate enough for practical usage. The next section describes the results of a set of experiments.

5. EXPERIMENTS

In this section, we provide a sampling of experimental results to show the magnitude of centroid errors and their effect on pose estimation. Since quantitative results like these are specific to a particular object and sensor configuration, we can only provide representative data to give an indication of the general magnitude of the errors.

As discussed earlier, two causes of centroid error are the surface tilt away from the image plane, and radial lens distortion. To show the effects of these two causes, we ran two separate simulations and varied these two parameters.

Figure 6 shows the effect of tilt angle on the centroid error. In this example, a circle was held fixed at a location along the optical axis of the camera, and it was tilted at angles ranging from 0° to 80° in the direction of the positive X-axis. The centroid errors in the image x-direction are plotted in Figure 6 as a function of tilt angle (the errors in the y-direction were zero). Four separate circle radii were tested, with the ratio of radius-to-range equal to 0.02, 0.06, 0.10, and 0.14. For the camera model, we used the same pinhole camera model used earlier, with "typical" camera parameters of focal length = 1 cm and pixel size = 0.0013 cm on a side. The results show that the centroid errors are significant for the circle radius of 0.06 and above, and also that they peak at an angle of 45°. To give an idea of the size of these circles, the circle of radius 0.06 would have a projected image radius of 46 pixels (at a tilt angle of 0°).

Figure 7 shows the effect of radial lens distortion on the centroid error. In this example, a circle with a radius-to-range ratio of 0.10 was moved horizontally parallel to the image plane, at a constant range Z_0. The pan and tilt angles were 0°; *i.e.*, the circle was parallel to the image plane. The circle was moved such that X_0 / Z_0 varied from 0.0 to 0.25, in increments of 0.05. The centroid errors in the image x-direction are plotted in Figure 7 as a function of X_0 / Z_0 (the errors in the y-direction were zero). The Tsai camera model was used, with four separate values for the lens distortion coefficient[3]: {–0.0009, –0.0018, –0.0049, –0.0112} (the values are unit-less). These values for radial lens distortion were actual values that we have obtained from calibrating real cameras and lenses in our lab. The focal length was again 1 cm and pixel size = 0.0013 cm on a side. The results show that the centroid errors increase almost linearly outward from the image

---

[3]Using Tsai's definition [Tsai88], the lens distortion coefficient κ relates distorted image points (X_d, Y_d) to undistorted points (X_u, Y_u) via the equations $(X_d, Y_d) = (2X_u/D, 2Y_u/D)$, where $D = 1 + (1 - 4\kappa R_u^2)^{1/2}$, and $R_u^2 = X_u^2 + Y_u^2$.

center, and at the edge of the image, they are relatively large for each of the tested values of lens distortion.

As a final example, we performed a simulation to test the effect of centroid error on pose estimation. We created a planar target consisting of four circle features arranged in a rectangle, measuring 5 cm by 4 cm. Each circle had a radius of 1 cm. This target was placed at a range of 10 cm and at an angle of 45° with respect to the camera. The camera had a focal length of 0.8 cm and a lens distortion of –0.0019. The projected circles fit completely inside the 560x480 pixel image. The centroid errors of the circles vary from about 1.7 pixels to about 4.1 pixels. The pose was computed from the image locations of the four target circles [Xu90]. The centroid errors cause the pose to have an error of about 0.33° and 0.02 cm. With the techniques described in this paper, these errors can be compensated for.

6. CONCLUSIONS

This paper has described a set of techniques for significantly improving the accuracy of image centroid features. Two specific causes of centroid errors are surface tilt and lens distortion. The techniques should be of interest to anyone that is interested in improving the accuracy of pose estimation and camera calibration results that are derived from image centroid features. We have described three methods: the first method, using a simulator, is easy to implement and can provide accurate (though slow) results. The second method, based on numerical integration, is faster but conceptually more complicated. The third method, using a neural network, is not as accurate but is fast enough for real time applications.

ACKNOWLEDGEMENTS

This work was performed with support from Martin Marietta Corporation Independent Research and Development projects D-11R and D-20R. We also thank J. Friedman for the MARS program.

REFERENCES

[Abid90] Abidi, M. and T. Chandra, "Pose Estimation for Camera Calibration and Landmark Tracking," *Proc. of IEEE Int'l Conference on Robotics and Automation*, 1990, pp. 420-426.

[Besl85] Besl, P. and R. Jain, "Three Dimensional Object Recognition," *Computing Surveys*, Vol. 17, No. 1, March 1985, pp. 75-145.

[Bose90] Bose, C. and I. Amir, "Design of Fiducials for Accurate Registration Using Machine Vision," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 12, No. 12, December 1990, pp. 1196-1200.

[Buck78] Buck, R., *Advanced Calculus*, McGraw Hill, New York, NY, 1978, pg. 391.

[Corn87] Cornils, K. and P. Goode, "Location of Planar Targets in Three Space from Monocular Images," *Proc. of 1987 Goddard Conf. on Space Applications of Artificial Intelligence and Robotics*, Greenbelt, Maryland, 1987.

[Davi87] Davis, V., "Systems Integration for the Kennedy Space Center Robotics Applications Development Laboratory," *Proc. of Robotic Systems in Aerospace Manufacturing Conf.*, Soc. of Manufacturing Engineers, September 1987, Fort Worth, Texas.

[DeBo78] DeBoor, C., *A Practical Guide to Splines*, Springer-Verlag, New York, 1978.

[Frie91a] Friedman, J., "Multivariate Adaptive Regression Splines," *Annals of Statistics*, March 1991, pp 1-141, 1991.

[Frie91b] Friedman, J., "Adaptive Spline Networks," *Advances in Neural Information Processing Systems 3*, Morgan-Kaufmann, San Mateo, 1991.

[Hara89] Haralick, R., H. Joo, C. Lee, X. Zhuang, V. Vaidya, and M. Kim, "Pose Estimation from Corresponding Point Data," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 19, No. 6, November-December 1989, pp. 1426-1446.

[Hoff91] Hoff, W., L. Gatrell, and J. Spofford, "Machine Vision Based Teleoperation Aid," *Telematics and Informatics*, Vol. 8, No. 4, pp. 403-423, Pergamon Press, December 1991.

[Hung85] Hung, Y., P. Yeh, and D. Harwood, "Passive Ranging to Known Planar Point Sets," *Proc. of IEEE Int'l Conference on Robotics and Automation*, 1985.

[Kris91] Krishnan, R., H. Sommer III, and P. Spidaliere, "Monocular Pose of a Rigid Body Using Point Landmarks," to appear in *Computer Vision, Graphics, and Image Processing*, 1991.

[Lane91] Lane, S., Handleman, D., and Gelfand, J., "Higher-Order CMAC Neural Networks: Theory and Practice," *Proceedings, 1991 IEEE American Control Conference*, Boston, 1991.

[Lape87] Lapedes, A., and Farber, R., "Nonlinear Signal Processing Using Neural Networks: Prediction and System Modeling," Los Alamos National Laboratory Technical Report, LA-UR-87-2662, 1987.

[Lenz89] Lenz, R. and R. Tsai, "Calibrating a Cartesian Robot with Eye-on-Hand Configuration Independent of Eye-to-Hand Relationship," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 11, No. 9, September 1989, pp. 916-928.

[O'Gor90] O'Gorman, L., A. Bruckstein, C. Bose, and I. Amir, "Subpixel Registration Using a Concentric Ring Fiducial," *Proc. 10th Int'l Conf. on Pattern Recognition*, IEEE Computer Society Press, June 1990, Atlantic City, New Jersey, pp. 249-253.

[Park91] Park, J. and Sandberg, J., "Universal Approximation Using Radial Basis Function Networks," *Neural Computation*, Vol 3, 246-257, 1991.

[Path89] Pathre, U., "Vision Based Automatic Theodolite for Robot Calibration," Ph.D. thesis, Dept. of Mechanical Engineering, Texas A&M University, 1989.

[Plat91] Platt, J., "A Resource-Allocating Network for Function Interpolation," *Neural Computation*, Vol 3, 213-225, 1991.

[Sand90] Sanderson, A., "Applications of Neural Networks in Robotics and Automation for Manufacturing," *Neural Networks for Control*, edited by W.T. Miller, R.S. Sutton, and P.J. Werbos, MIT Press, 1990.

[Skla90] Sklair, C., Gatrell, L., Hoff, W., and Magee, M., "Optical Target Location Using Machine Vision in Space Robotics Tasks," *Proceedings, Cooperative Intelligent Robotics in Space*, SPIE Vol. 1387, November 1990.

[Skla91] Sklair, C., W. Hoff, and L. Gatrell, "Accuracy of Locating Circular Features Using Machine Vision," *Proceedings, Cooperative Intelligent Robotics in Space*, SPIE Vol. 1612, November 1991.

[Tsai87] Tsai, R., "A Versatile Camera Calibration Technique for High Accuracy 3D Machine Vision Metrology Using Off-the-shelf TV Cameras and Lenses," *IEEE Journal of Robotics and Automation*, Vol. RA-3, No. 4, August 1987, pp. 323-344.

[Tsai88] Tsai, R. and R. Lenz, "Overview of a Unified Calibration Trio for Robot Eye, Eye-to-Hand, and Hand Calibration using 3D Machine Vision," *Proc. of SPIE Conference on Sensor Fusion: Spatial Reasoning and Scene Interpretation*, Vol. 1003, 1988, pp. 202-213.

[Tsai89] Tsai, R. and R. Lenz, "A New Technique for Fully Autonomous and Efficient 3D Robotics Hand/Eye Calibration," *IEEE Transactions on Robotics and Automation*, Vol. 5, No. 3, June 1989, pp. 345-358.

[Xu90] Xu, W., "Optimal 3-D Motion Estimation From Images of Feature Points," Ph.D. thesis, University of Colorado at Boulder, Dept. of Electrical and Computer Engineering, 1990.

[Yuan89] Yuan, J., "A General Photogrammetric Method for Determining Object Position and Orientation," *IEEE Transactions on Robotics and Automation*, Vol. 5, No. 2, April 1989, pp. 129-142.

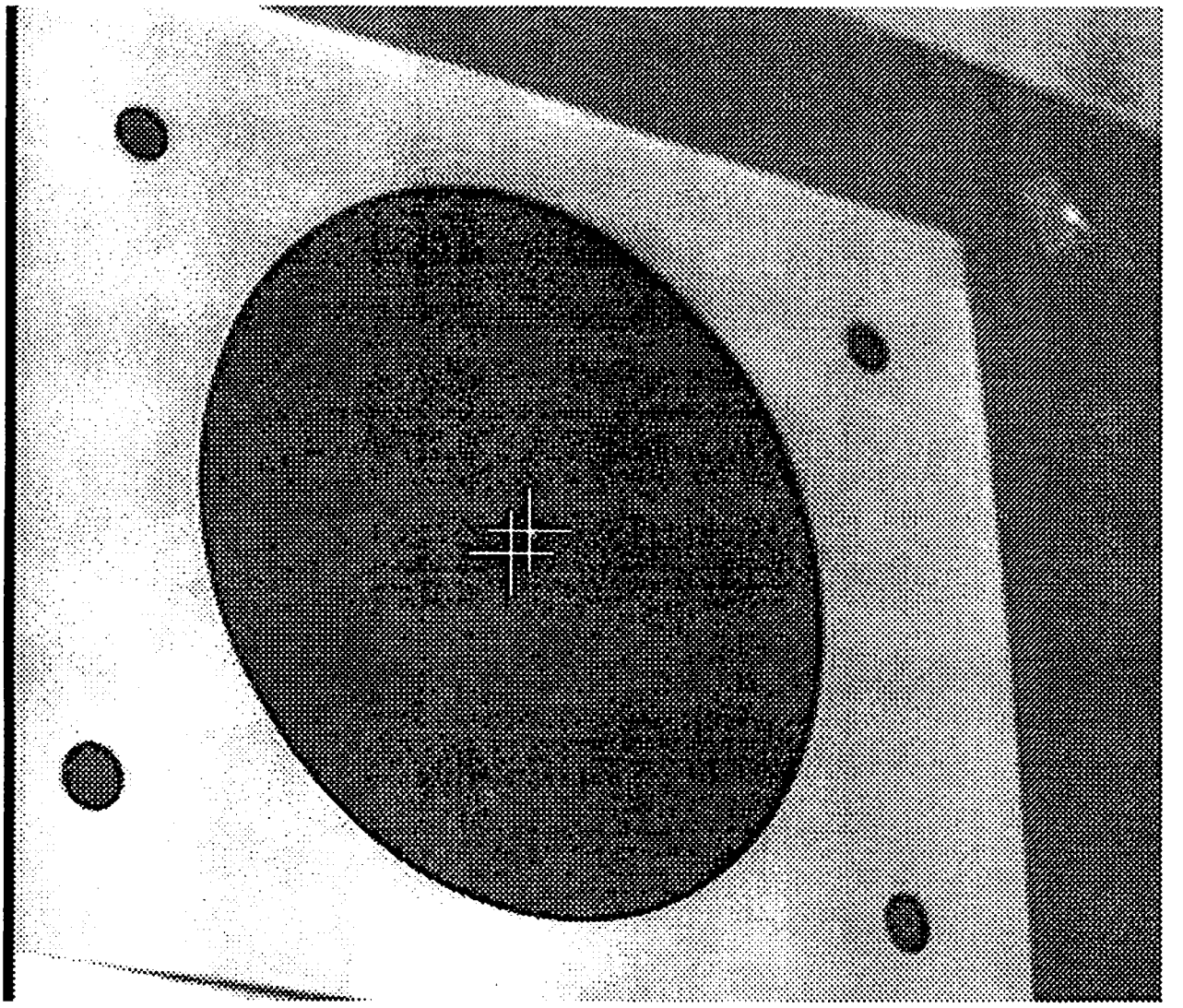

Figure 1. An actual digitized image showing the error between the computed image centroid (left crosshair) and the true projected center of the circle (small white dot under right crosshair). The right crosshair marks the corrected centroid location.

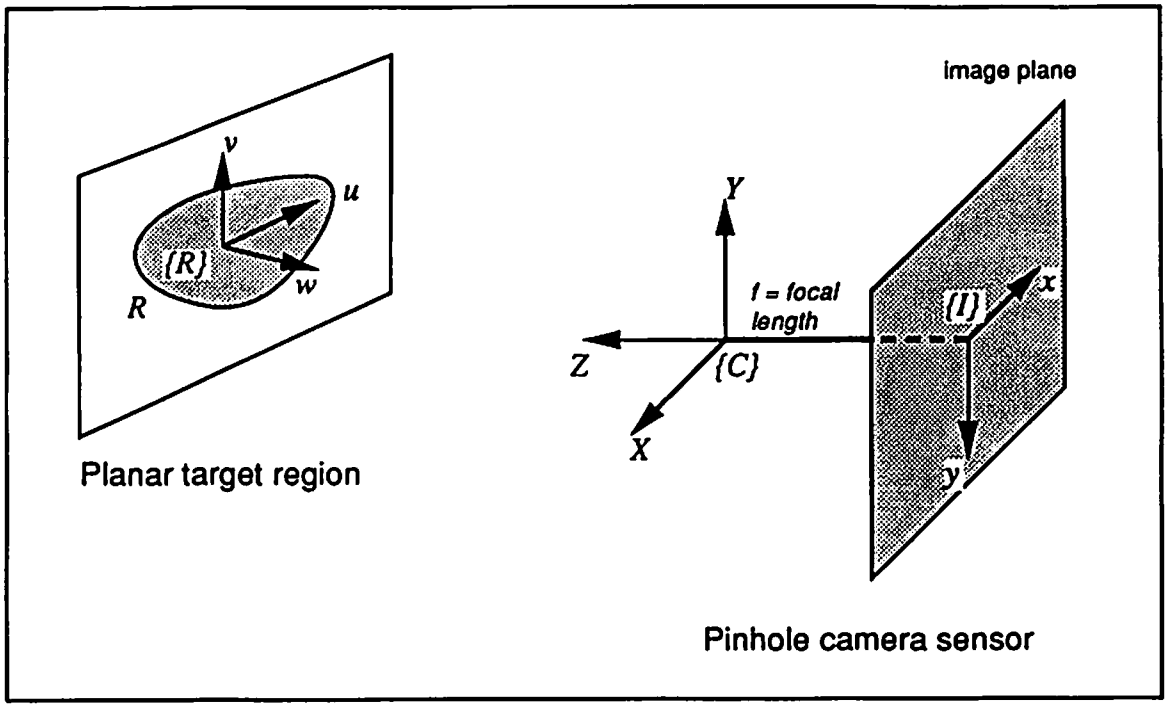

Figure 2. Projection of a planar fiducial region R onto the image plane using a pinhole camera model.

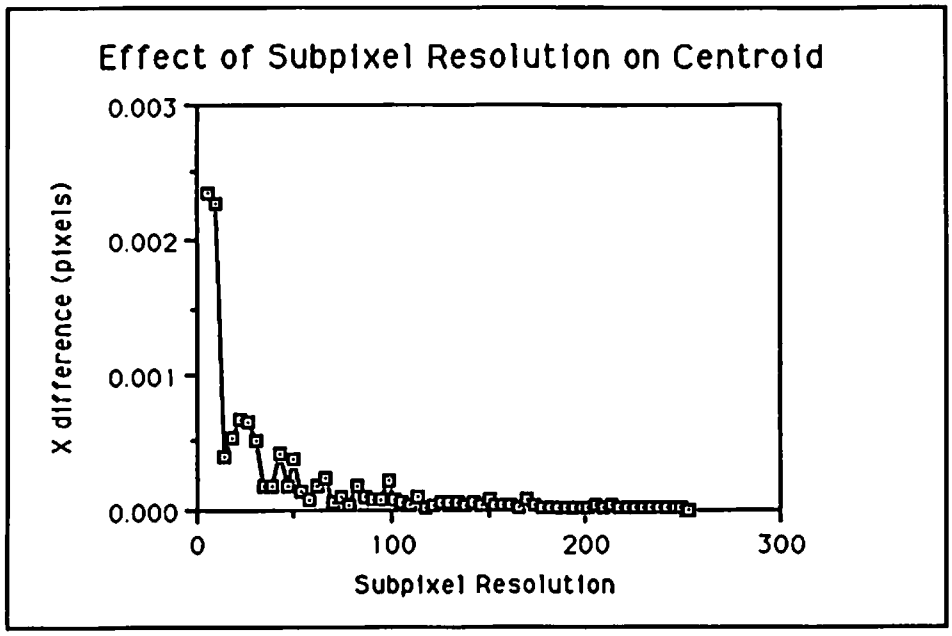

Figure 3. Maximum difference between the centroid computed using subpixel resolution 254x254 and centroids computed using coarser resolutions, using the simulator.

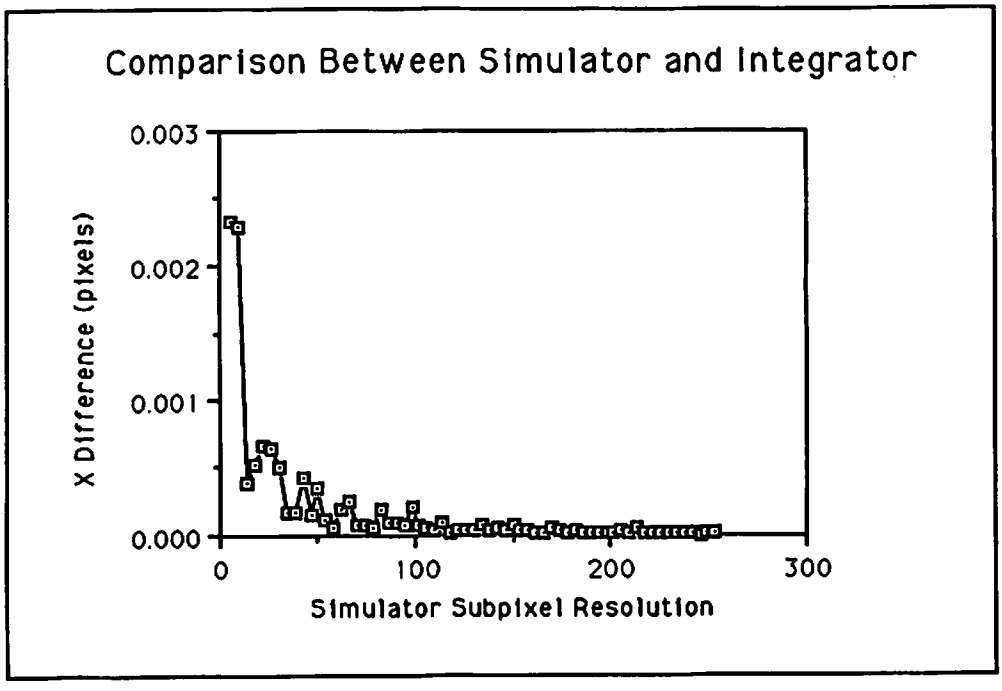

Figure 4. Maximum difference between centroids computed using the simulator, at varying subpixel resolutions, and the centroids computed using the numerical integrator.

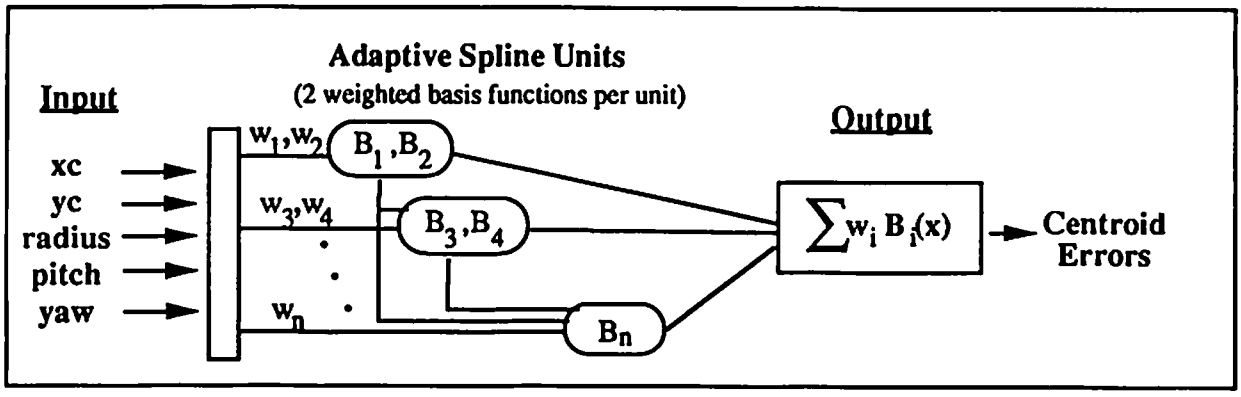

Figure 5. Adaptive spline neural network for predicting centroid error.

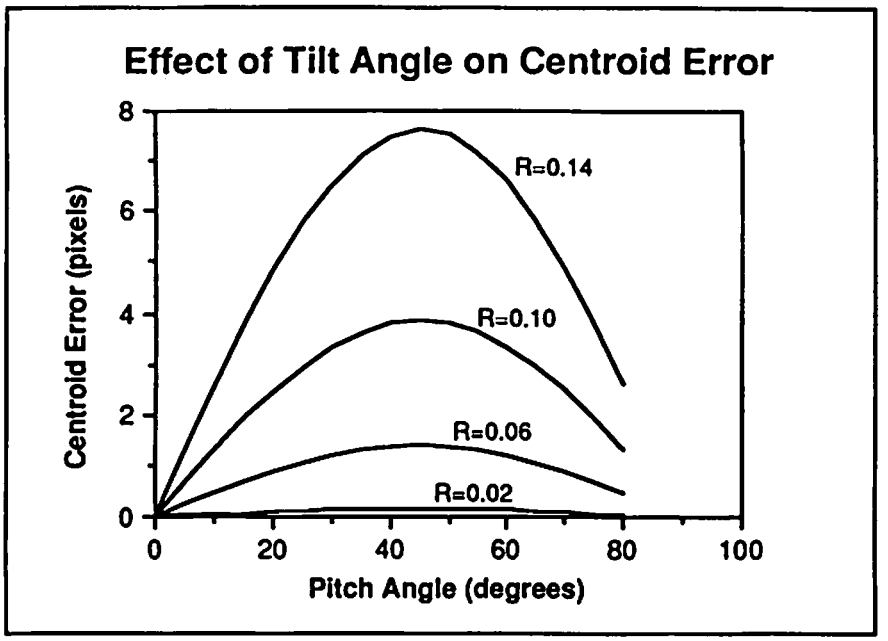

Figure 6. Effect of surface tilt away from the image plane, for a set of circle fiducials with fixed radii, located at a fixed distance along the camera's optical axis.

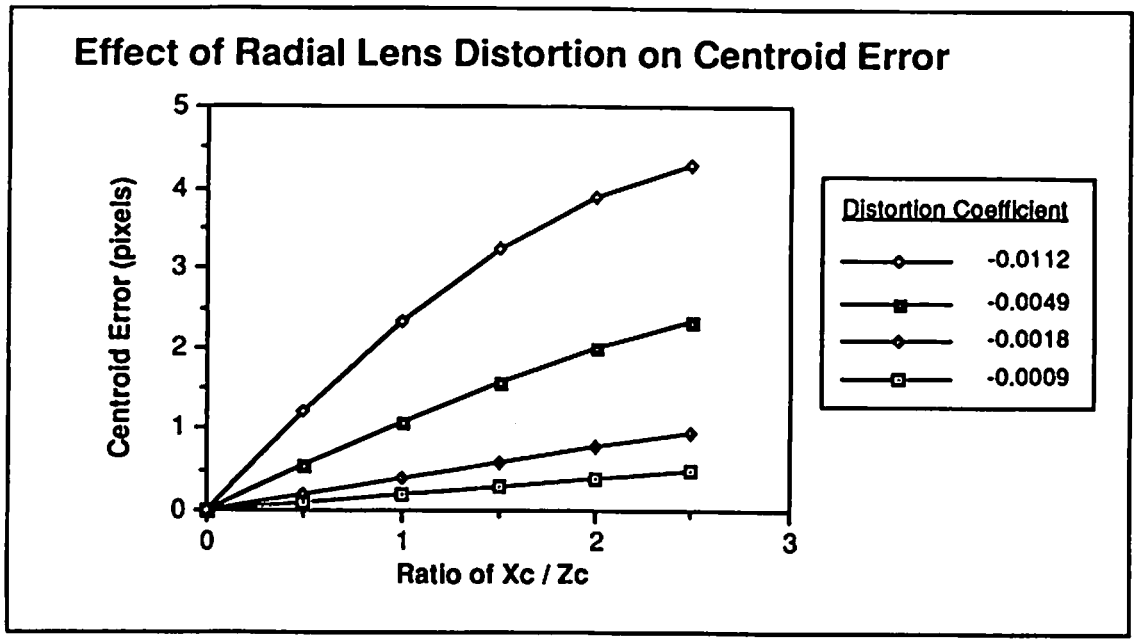

Figure 7. Effect of different values of radial lens distortion, for different locations of a circle fiducial.