Position Verification of a Mobile Robot Using Standard Pattern

MANSUR R. KABUKA, MEMBER, IEEE, AND ALVARO E. ARENAS

Abstract—**As mobile robots are taking on more and more of the tasks that were normally delegated to humans, they need to acquire higher degrees of autonomous operation, which calls for accurate and efficient position determination and/or verification. The critical geometric dimensions of a standard pattern are used here to locate the relative position of the mobile robot with respect to the pattern; by doing so, the method does not depend on values of any intrinsic camera parameters, except the focal length. In addition, this method has the advantages of simplicity and flexibility. This standard pattern is also provided with a unique identification code, using bar codes, that enables the system to find the absolute location of the pattern. These bar codes also assist in the scanning algorithms to locate the pattern in the environment. A thorough error analysis and experimental results obtained through software simulation are presented, as well as the current direction of our work.**

I. Introduction

AS RESEARCH has developed on mobile autonomous robots, accurate position determination relative to some standard coordinate system has become necessary. Various methods have been proposed to accomplish this; these methods involve the use of buried wires [1], painted lines [2], [3], infrared beacons [4], ultrasonics [5], and sophisticated three-dimensional viewing systems [6]. The basic problem is to develop a simple noninvasive inexpensive system which can be easily implemented in a wide variety of manufacturing environments.

It is our contention that the vast majority of robotics guidance applications will not require a high degree of accuracy while the robot is in motion but that most of the high-accuracy work will be done after the robot locates itself near a particular operating station. In this context, then, normal motion of the robot could be done by a simple trajectory planning system based on time measurements at a constant velocity or other similar methods dependent on mechanical data of the motor portion of the robot. Once the robot is positioned close to its desired working station, at the end of its planned trajectory an accurate position verification based on direct visual measurements can be used to appropriately correct the inaccuracies inherent in mechanical measurements. It is the latter part, accurate visual position verification, with which this paper is concerned.

Towards this purpose, we propose the use of a standard pattern such that the position of the robot with respect to this pattern can be determined by direct observation of a single projection and minimal knowledge of intrinsic camera parameters. Use of such standard marks has been proposed elsewhere [7], [8]; the basic idea is that the working station at which the robot must be positioned is at a known position with respect to the pattern.

Given the nature of the system we propose, it is possible that the robot will end up in a position from which it cannot view the pattern; thus an appropriate method of scanning is necessary. The main objective in developing a specific scanning algorithm is to reduce the time required for the camera to view the pattern.

Most of the work in standard patterns has implicitly concentrated on an environment with a single working station. However, it is likely that in many tasks the robot would need to hover from one station to another; moreover, the existence of several patterns in the working environment allows for recovery if the robot is absolutely lost, if it is assumed that the location of each pattern in some standard world coordinate system is known. Furthermore, the added flexibility of having several patterns can be used in a navigation system where the robot could be instructed to follow a sequence of patterns. Thus we intend to associate a unique code with each pattern; this will enable a system to identify the location of the pattern in the world coordinates.

II. Description of the System

The pattern will consist of two main areas, as illustrated in Fig. 1: the relative displacement pattern and the identification codes. The relative displacement pattern will be used to obtain the relative position of the viewing point with respect to the location of the pattern by analysis of the particular geometric characteristics of its projection onto the image plane. The identification codes will serve two purposes: they will provide a unique code such that the viewed pattern can be adequately discerned from other patterns in the environment, and they will also be an aid to scan for the pattern in a minimal amount of time.

We have sought a displacement pattern such that when viewed at any given position, the geometric characteristics of its projected image uniquely determine the said position. We have chosen to use a circle since its resultant projection on the image plane, assuming no projective distortion, always results in an ellipse whose size, eccentricity, and orientation can be used to determine the viewing position. To specify the ellipse, we will use the semimajor axis b, the semiminor axis a, and the angle ψ that the semiminor axis makes with the vertical, as

Manuscript received March 3, 1986; revised June 1, 1987.

M. R. Kabuka is with the Department of Electrical and Computer Engineering, P. O. Box 248294, University of Miami, Coral Gables, FL 33124.

A. E. Arenas is at 6514 S.W. 107th Place, Miami, FL 33173.

IEEE Log Number 8716959.

0882-4967/87/1200-0505$01.00

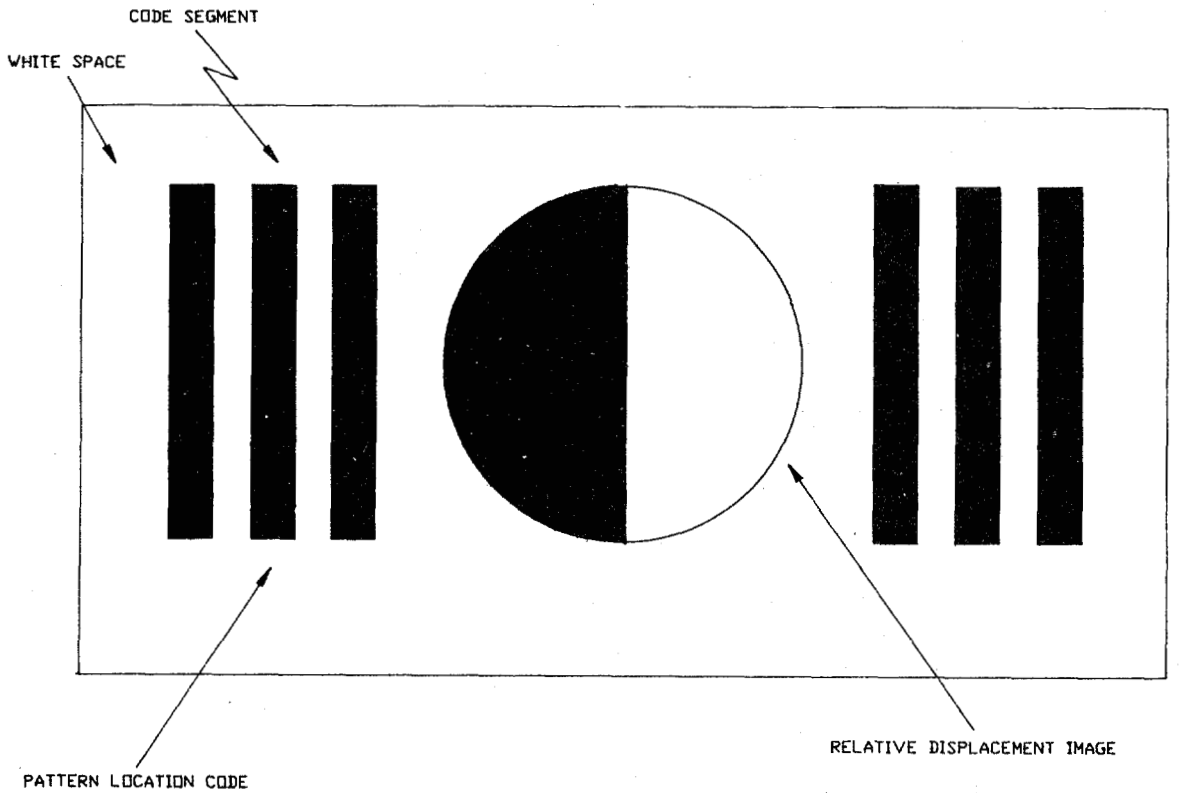


Fig. 1. Simplest implementation of proposed pattern. Position verification is possible only when viewed from one of two possible viewing quadrants.

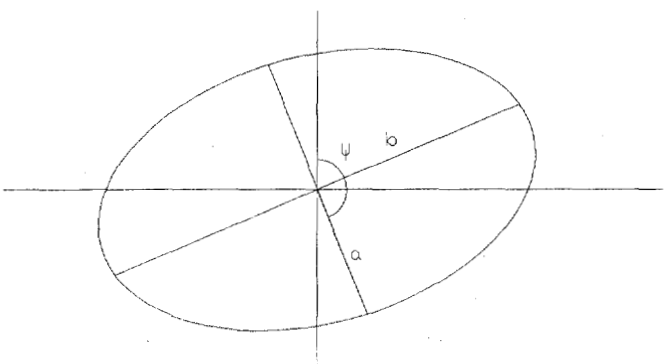


Fig. 2. Ellipse parameters.

illustrated in Fig. 2. Obviously, an error is introduced due to the projective transformation; that is, the projected image is not quite an ellipse. We shall note later, however, that this error is negligible.

An additional consideration that prompted us to use a circle was its ease of implementation. As such, it can be discretely placed on one wall of the environment without imposing a heavy burden, as a three-dimensional pattern would. The circle itself is quite simple to specify since only the dimension of its radius is needed.

The portion of the pattern dedicated to identification will be designed based on the concept of start/stop codes in bar code schemes [9]. It is quite simple to design the bar codes such that they will be readable from almost any distance or angle of view; such designs include the use of relative distances between bars, binary coding on the bars (e.g., thick and thin), or even by utilizing color codes. In addition, the nature of the bar codes allows us to use them for rapid location of the pattern in the field of view since even if they are only partially visible, they can be appropriately identified and decoded.

The next two sections will be devoted to the calculations regarding the relative displacement of the camera from the pattern, assuming the displacement pattern centered in the field of view. Discussion on the bar code schemes will be deferred until the section on scanning and identification.

III. Determination of the Relative Displacement

Consider the orthogonal coordinate system $(\hat{x}, \hat{y}, \hat{z})$ with its origin at the center of the displacement pattern and such that $\hat{z}$ is perpendicular to the wall. Let us now define a second coordinate system $(\hat{x}', \hat{y}', \hat{z}')$ at a distance f from the lens center, where f is the focal length of the camera, and the coordinate is set such that the lens center lies on the $\hat{z}'$ axis. The plane determined by the $\hat{x}'$ and $\hat{y}'$ axes is then the image plane, and its coordinates are set such that the projection of the $\hat{y}$ axis on the image plane is always the $\hat{y}'$ axis, i.e., it is assumed that the swing angle of the camera is zero.

We will define the position of the lens center with respect to the center of the pattern by the spherical coordinates: distance D, elevation angle θ, and co-azimuth angle β, as illustrated in Fig. 3. We will call β the co-azimuth angle because it is referred to the vertical instead of to the horizontal; this choice is natural noting the previous assumption on the swing angle.

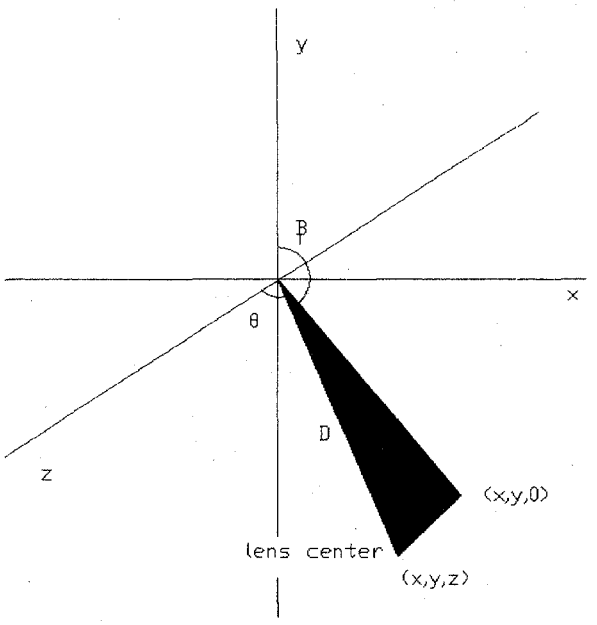


Fig. 3. Spherical coordinate representation.

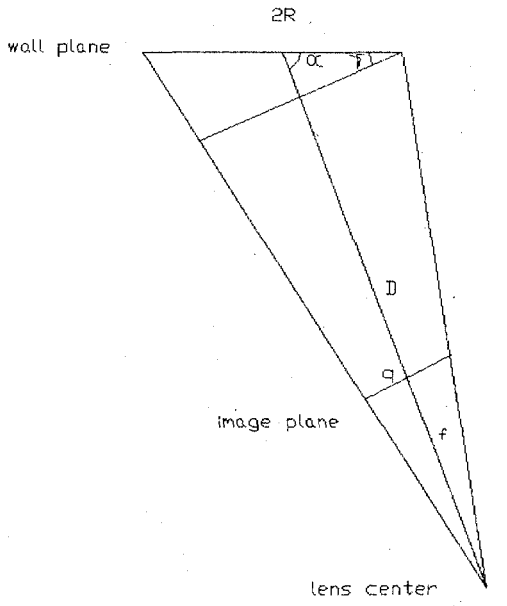


Fig. 4. Generation of elliptical axis q.

Since the pattern is assumed affixed to a wall,

$$0 \le \theta \le 90°; \tag{1}$$

however, the co-azimuth angle β is defined over the whole range

$$-180° \le \beta \le 180°. \tag{2}$$

We will specify the position of the pattern in spherical coordinates because it is easier to calculate these from the measurable parameters in the ellipse. From these obtained parameters a transformation into the usual Cartesian coordinates can be easily made; however, it is contended that a specification in spherical coordinates is, in general, more adequate that the usual Cartesian coordinates for most tasks since it directly defines the relative distance and the angle of view, which are in turn more directly related to the motion parameters of the camera mount.

Let us assume for now that the viewed ellipse is centered in the image plane; discussion on the issues of scanning will be done later on. If this is the case, the optical axis of the camera passes through the center of the circle. Each diameter will generate an axis q in the ellipse, as shown in Fig. 4, depending on the angle α formed from the generating diameter to the optical axis, such that defining

$$\gamma = 90° - \alpha, \tag{3}$$

we obtain

$$q = \frac{2fDR \cos \gamma}{D^2 - R^2 \sin^2 \gamma}. \tag{4}$$

It is known that a unique diameter exists perpendicular to the optical axis, that is, $\alpha = 90° \rightarrow \gamma = 0°$. This diameter generates the semimajor axis

$$b = \left(\frac{f}{D}\right) R; \tag{5}$$

therefore, given b, we can calculate the distance as

$$D = (fR)\frac{1}{b}. \tag{6}$$

The semiminor axis is generated perpendicular to the semimajor axis; this is the situation where α is minimum. This diameter lies on the projection of the optical axis on the wall plane, and $\gamma = \theta$ for this condition. Thus replacing into (4),

$$a = \left(\frac{fDR \cos \theta}{D^2 - R^2 \sin^2 \theta}\right). \tag{7}$$

This equation can be used to find the angle θ since the distance D can be calculated as shown before. However, it can be seen that it is not simple to solve for θ in this equation. Instead, it is possible to assume that

$$R \sin \theta \ll D. \tag{8}$$

This assumption is equivalent to stating that the projected pattern is a true ellipse; in reality, the projected minor axis is slightly longer at the farther half. At very small distances, in the order of magnitude of the radius of the circle, the error due to this assumption would be quite strong; however, we believe that the pattern would not need to be seen at such small distances. In any case, the validity of the assumption is stronger at small angles; thus it is recommended that the working station be positioned as close to $\theta = 0$ as possible. The equation for the semiminor axis thus simplifies to

$$a \approx \left(\frac{f}{D}\right) R \cos \theta, \tag{9}$$

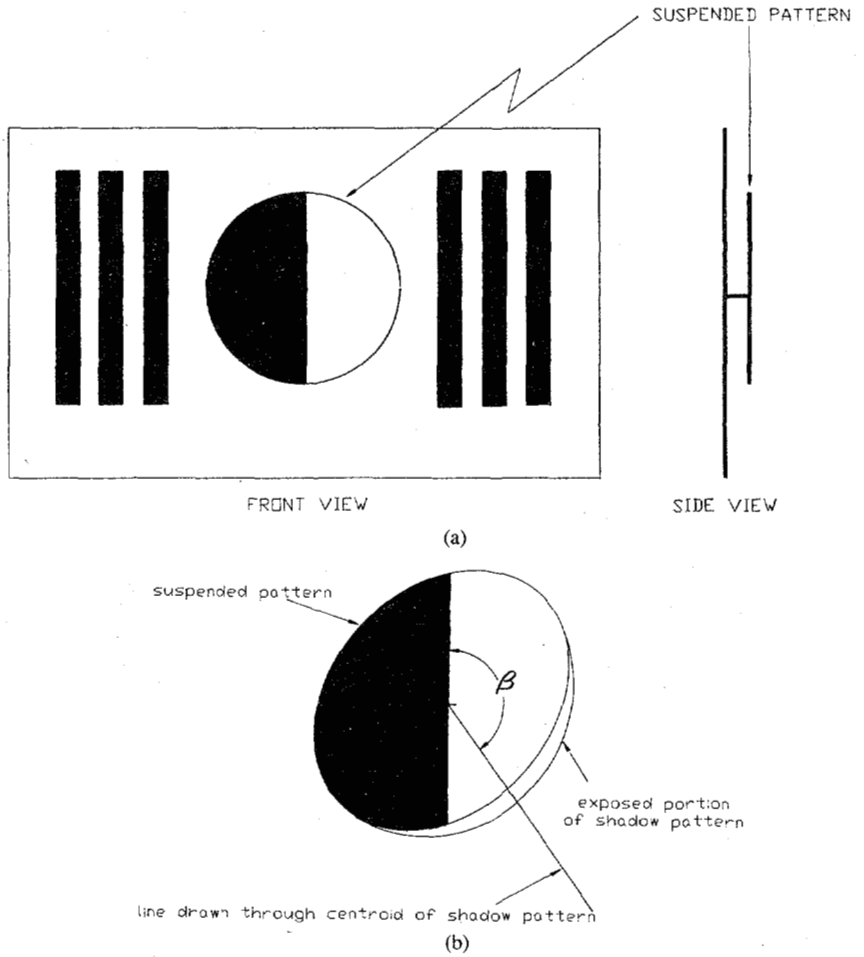


Fig. 5. (a) More complex version of proposed pattern. Position calculation is possible from any viewing. (b) Pattern viewed at co-azimuth angle β.

and it is then easily shown that

$$\theta = \arccos\left(\frac{a}{b}\right). \tag{10}$$

Furthermore, if we define the *expected radius* R' such that

$$R' = \frac{f}{D} R \tag{11}$$

would be the radius of the observed circle if the image plane is at the same distance but parallel to the wall, it is possible to manipulate (10) so that

$$\theta = \arccos\left(\frac{\pi ab}{\pi R'^2}\right) = \arccos\frac{A_e}{A_c} \tag{12}$$

where A_e is the area of the ellipse, and A_c is the area of the expected circle. In this way, it is possible to calculate θ by a simple pixel count on the image.

The orientation angle ψ is obtained by noting that the semiminor axis is generated by the projection of the optical axis; since the plane determined by this projection also contains the $\hat{z}$ axis of the wall coordinate system,

$$\psi = \begin{cases} \beta, & \text{if } \beta \geq 0°; \\ \beta - \pi, & \text{otherwise.} \end{cases} \tag{13}$$

From this condition we obtain that

$$\psi = \beta + l\pi \tag{14}$$

where l is either 0 or 1, and thus there remains a two-way ambiguity. In some applications it may not be a big constraint to limit β from 0 to 180°, which is equivalent to limiting the position of the viewing point to a quarter space. For example, it can be assumed that all patterns lie above the plane of view of the camera. However, in the general case such an assumption is unacceptable, and thus we must solve this ambiguity problem.

One simple way of doing so is to make the pattern suspended, that is, at a certain distance T from the wall. Fig. 5(a) shows this arrangement, consisting of a "suspended pattern" and its respective "shadow pattern"; the suspended pattern is identical to the circular image found in Fig. 1. The way the shadow will be viewed when the ellipse is tilted at an angle β with the vertical is shown in Fig. 5(b). The same shadow effect can be obtained by using a cylinder of a certain thickness T in place of a circle; the "shadow" would then be

the side of the cylinder. This second method is perhaps better suited for an industrial environment since if T is small enough it can be affixed to a wall without elaborates support; actually suspending the pattern may cause errors due to motion of the pattern caused by winds or other environmental conditions. In either case, the shadow would have to be of a different color to facilitate distinction; once the position of the shadow in the image plane is identified, the parameter l in (14) can be easily determined by observing at which side of the major axis the shadow lies. Since this observation is of a qualitative nature only, small errors in calculation would not affect it at all.

We must determine an appropriate thickness T such that the shadow is visible from a desirable range. It must not be too small as to not be detectable at a maximum specified distance, while it must not be so large that it obscures vital information (such as the bar codes) or that it takes up so much of the image that the pattern cannot be seen. Let us define t to be the thickness of the shadow image in the line determined by the minor axis; t can be shown to be approximately

$$t = \frac{Tf \sin \theta}{D}. \tag{15}$$

Now, let us define $D_{\max}$ to be the maximum distance at which the pattern must be viewed and $\Delta\theta$ the minimum angle; assuming that $N_{t_{\min}}$ is the minimum number of pixels desirable for t at such a distance and angle, we find that the thickness of the pattern must be at least

$$T_{\min} \geq \frac{N_{t_{\min}} \Delta x D_{\max}}{f \sin \Delta\theta} \tag{16}$$

where Δx is the length of the pixels in the x direction. The largest appropriate thickness will be given by a similar equation, defining a minimum distance and a maximum angle.

IV. Calculation of the Elliptical Parameters

Up to this point we have discussed how to obtain the coordinate location of the lens center using the values of the parameters of the viewed ellipse, implicitly assuming that these parameters are known. In this section, we will outline the procedure of computation of these parameters.

Let us initially consider the simplified case where $\beta = \pm 90°$, which is achieved by fixing the camera and the center of the pattern to be at the same height. This assumption forces the major and minor axes to be vertical and horizontal, respectively, in the image plane. Thus the major and minor axes become the length of the vertical and horizontal projections of the ellipse, respectively, and they are easily computable in real time. The angle ψ is completely determined in the assumption, and, therefore, it does not need to be calculated. The ambiguity in β, however, must be resolved, and this is done using the shadow pattern method outlined before. As such, the projection of the shadow in the x direction completely specifies the sign of the angle, perhaps by a weighted sum calculation. Based upon this simplified case, we have performed several experiments, the results of which are outlined in a later section.

The restriction on the heights of the sensors may be acceptable in many applications, especially in highly constrained environments. However, as will be seen later, even small changes in height may cause unsolvable problems in the computation of the relative displacement; therefore, a more general solution seems necessary. To calculate the size and orientation of the image, we can make use of the lower order position-invariant moments of the ellipse:

$$b = \left(\frac{\mu_{20} + \mu_{02} + [(\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2]^{1/2}}{\mu_{00}/2} \right)^{1/2} \tag{17}$$

$$a = \left(\frac{\mu_{20} + \mu_{02} - [(\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2]^{1/2}}{\mu_{00}/2} \right)^{1/2} \tag{18}$$

$$\psi = 90° + \frac{1}{2} \arctan \left(\frac{2\mu_{11}}{\mu_{02} - \mu_{20}} \right) \tag{19}$$

where b is the semimajor axis, a is the semiminor axis, and ψ is the angle made by the semiminor axis with the vertical. Although this seems to be a highly computationally intensive solution, remember that all these computations need to be done only once per frame, assuming that the moments can be obtained at this rate. Thus the problem arises in computing the moments themselves.

The equation defining the moments $m_{a,b}$ of an $M \times N$ intensity array $I(i, j)$ is

$$m_{a,b} = \sum_{i,j} I(i, j) i^a j^b \tag{20}$$

where $a + b$ is the order of the moment, i is the column, and j is the row. By defining one-dimensional arrays $I(t)$ where $t = i + M \times j$, we can consider moment generation as the dot product

$$m_{a,b} = \langle I(i, j), p_{a,b}(i, j) \rangle \tag{21}$$

where

$$p_{a,b}(i, j) = i^a j^b \tag{22}$$

is made one dimensional in the same ways as I. Andersson [10] proposed a low-order moment generator chip using only counters, shift registers, and accumulators. Obviously, zero- and first-order moments only require accumulators and counters; however, it seems that a second-order moment generator would require a multiplier. This is not the case, however; we can perform an iterative computation of the factors $p_{a,b}$, *where* $a + b = 2$, as follows:

$$p_{2,0}(i+1, j) = p_{2,0}(i, j) + 2i + 1 \tag{23}$$

$$p_{1,1}(i+1, j) = p_{1,1}(i, j) + j \tag{24}$$

$$p_{0,2}(i, j+1) = p_{0,2}(i, j) + 2j + 1 \tag{25}$$

with special cases for top of screen and left margin.

By decomposing $I(i, j)$ as

$$I(i, j) = 2^7 I(i, j)_7 + 2^6 I(i, j)_6 + \cdots + I(i, j)_0, \tag{26}$$

we can rewrite the moment equation as

$$m_{a,b} = 2^7 F_7 + 2^6 F_6 + \cdots + F_0 \tag{27}$$

where

$$F_k = \langle I(i, j)_k, p_{a,b}(i, j) \rangle. \tag{28}$$

The F_k require only $1 \times n$ bit multiplication, easily achieved by a set of n AND gates, where n is the number of bits in p. $m_{a,b}$ needs to be evaluated only once per frame and can be done by any method, such as Horner's method of polynomial evaluation. Thus the moments can be easily calculated in real time.

The position-invariant moments μ are defined in terms of these by equating $\mu_{1,0} = \mu_{0,1} = 0$ and adjusting all others as follows:

$$\mu_{0,0} = m_{0,0} \tag{29}$$

$$\mu_{2,0} = m_{2,0} - \frac{m_{1,0}^2}{m_{0,0}} \tag{30}$$

$$\mu_{1,1} = m_{1,1} - \frac{m_{1,0} m_{0,1}}{m_{0,0}} \tag{31}$$

$$\mu_{0,2} = m_{0,2} - \frac{m_{0,1}^2}{m_{0,0}}. \tag{32}$$

Then the values obtained for $\mu_{a,b}$ can be used in (17)–(19) to calculate the elliptical parameters.

V. Scanning and Identification

We have assumed up to this point that the pattern image has been centered on the image plane; let us discuss now the issue of scanning for the pattern. Once the robot completes its predetermined trajectory, it must locate the pattern within its field of view. Instead of searching for the elliptical pattern, however, scanning is made simpler by instructing the robot to search for the bar codes; a partially seen ellipse cannot be identified as an ellipse, while partially viewed bar codes are still bar codes. An additional advantage of scanning for the bar codes instead of looking directly for the pattern is that identification of the pattern must be done previous to any sort of position verification and adjustment lest the robot waste time in adjusting its position relative to the wrong pattern.

The bar codes, then, would act as pointers towards the actual location of the pattern itself and can easily be used to appropriately center the pattern. For this purpose the bar codes are placed vertically at both sides of the circular pattern, and it is known that the circle will lie in between. The bar codes must be larger than the diameter of the circle since identification of the pattern needs to be done even if the optical axis of the camera is at an angle with the wall.

If, due to motion inaccuracies, the pattern is not within the field of view of the camera, the camera must be rotated until the pattern is seen. Initially, let us assume that the relative height of the pattern with respect to the viewing point is known; in this way, the search for the pattern is only one dimensional. This assumption is not too constraining since, in general, the height of the pattern and the height of the camera with respect to the floor can be known in advance, and even if there is a small displacement Δz in the height of the camera, the pattern would be at least partially visible. The simplest way to search for the pattern would then be to turn the camera in a certain direction until the pattern is found.

The rate at which the camera rotates is critical. It should not be so fast as to cause missing the pattern, and it should not be so slow as to make the scanning too time consuming. The rate of rotation and the various system parameters are related by

$$\frac{\phi_v - \phi_0}{360^\circ} = \frac{\rho_r}{\rho_v} \tag{33}$$

where

ϕ_v angle covered by the field of view of the camera,
ϕ_0 angle at which successive views overlap,
ρ_r rate of rotation,
ρ_v video rate.

Normally, ϕ_v and ρ_v are fixed by the system, and thus $\rho_{r_{\max}}$ is determined by the overlap required (i.e., ϕ_0). The angle of overlap ϕ_0 is chosen such that the whole pattern appears in at least one frame. The worst case occurs when a little part of it is still out of view. Therefore, the pattern must be recognized in the next frame. For the whole pattern to fit in the next frame, ϕ_0 must be as large as the maximum angle of view that the pattern takes up. The maximum angle is given by the minimum distance $D_{\min}$ at which the pattern must be recognized and the radius R of the pattern as

$$\phi_{0_{\max}} = 2 \arctan\left(\frac{R}{D_{\min}}\right). \tag{34}$$

Then the maximum rate of camera rotation is given by

$$\rho_{r_{\max}} = \frac{\phi_v - \phi_{0_{\max}}}{360^\circ} \rho_v. \tag{35}$$

If the robot is moving, a further effect is caused by the angle the pattern moves in the field of view of the camera. The effect is maximum when the pattern is closest to the robot and when the motion is perpendicular to the camera. The maximum angle moved caused by the motion of the robot is given by

$$\phi_{m_{\max}} = \arctan \frac{v_{\max}}{\rho_v D_{\min}} \tag{36}$$

where

$v_{\max}$ maximum velocity of robot,
D_{min} minimum distance between camera and pattern,
ρ_v video rate.

Thus

$$\rho_{r_{\max}} = \frac{\phi_v - \phi_{0_{\max}} \phi_{m_{\max}}}{360^\circ} \rho_v. \tag{37}$$

In most of the cases, the effect of ϕ_m is small and hence can be neglected.

If we remove the restriction on one-dimensional searching,

then horizontal scanning of the pattern would not be enough. We suggest that the search be performed in a helical motion, by varying the horizontal and vertical angles independently in a sinusoidal fashion with linear increases in amplitude. The linear amplitude variation at each angle must obey the constraints stated earlier over each 360° rotation. Such a helical motion is deemed best since the pattern is assumed to be close to the robot, but no indication exists as to the direction in which the robot should start searching.

Once the bar codes are detected within the field of view, the control is modified so that motion is made linear in the direction pointed to by the bar codes. The start/stop codes would then be detected, and a consultation on its database would allow the robot to determine if this is the correct pattern. If it is not, search resumes for a new pattern, with the added information of the relative position between the desired and the observed pattern; otherwise, the pattern would be appropriately centered in the image plane, and position verification would commence.

The bar codes can also be helpful in the overall functioning of the system; specifically, since the bar codes also have certain dimensions associated with them, these can be used for double checking or averaging with the values obtained from the geometric calculations on the ellipse. We shall not address this issue in more detail, but its potential is quite apparent. In addition, since it is known that the patterns are vertically oriented, they can be quite useful for on-line correction of the swing angle of the camera; errors due to this fact would then be greatly minimized.

VI. Sources of Error

A. Simplified Case

As has been done throughout the paper, we will first assume the pattern and the camera to be at the same height, that is, the tilt angle β is fixed at $\pm 90°$; recall that in this case the elliptical parameters a and b are obtained by pixel counts on the projections of the segmented image. Errors are introduced into our calculations due to sensor resolution and noise, swing angle of the sensor, and sensor height variations. We shall analyze each of these conditions separately, as they affect the values of the calculated parameters D and θ. These conditions suggest the optimum values for R, the radius of the circular pattern, and the best location of the patterns with respect to the working stations they represent.

1) Sensor Resolution and Noise Errors: A number of factors contribute a certain amount of error in the calculated values for a and b; the overall effect of all these contributions is an uncertainty of $\pm k$ pixels in the measured values. Some of the factors that affect the method in this way, and possible solutions to them, are the following.

1) The resolution of the sensing element is the ultimate limiting factor in any image processing system; even if all other sources of error can be avoided completely, this still introduces an uncertainty of ± 1 pixels in any computation [2].

2) The segmentation of the image is extremely critical. In our application we can assume to have complete control over the environment, and thus segmentation can be enhanced by choosing the pattern distinct enough from the environment. A good solution to the segmentation problem is to use a color camera with RGB (red, green, blue) outputs, perhaps handling the segmentation in the analog signals themselves. Such a technique offers several potential advantages, including increased speed, improved accuracy, and ease of implementation.

3) Correct focusing of the image is important since an out-of-focus pattern will contain blurred edges that could introduce some errors upon segmentation. This can be easily handled by using autofocusing cameras.

4) The glare of the material from which the pattern is constructed can also introduce some errors. In real-life situations, this would require that the pattern be constructed of materials which are designed to attenuate glare.

Analysis of the errors associated with such a $\pm k$-pixel uncertainty in the computed values has been done by Drake *et al.* [2]; they have shown that the maximum percentage error associated with a pixel count calculation of this sort is

$$|e| = 100 \frac{k}{N} \tag{38}$$

where N is the number of pixels counted for the parameter. Let us analyze the effect of these errors in each of our computed values.

The value obtained for the distance D is given by (6). Let N_b be the number of pixels computed for the major axis $2b$, and let Δy be the length of a pixel in the y direction. Then

$$2b = N_b \Delta y, \tag{39}$$

and thus

$$D = \frac{2fR}{N_b \Delta y}. \tag{40}$$

Due to pixel uncertainty, the range of calculation for the distance would be

$$\bar{D} = \frac{2fR}{(N_b \pm k)\Delta y} \tag{41}$$

and the maximum error associated with this uncertainty is

$$\begin{aligned} |E|_D &= |D - \bar{D}| \\ &= \frac{2fR}{\Delta y} \left| \frac{1}{N_b} - \frac{1}{N_b \pm k} \right| \\ &= \frac{2fRk}{\Delta y N_b (N_b \pm k)}. \end{aligned} \tag{42}$$

The percentage error is then given by

$$\begin{aligned} |e| &= 100 \frac{|E|}{D} \\ &= 100 \left(\frac{k}{N_b \pm k} \right). \end{aligned} \tag{43}$$

If we assume $k \ll N_b$, then this equation simplifies to

$$|e| = 100 \frac{k}{N_b} \tag{44}$$

which, as can be seen from (38), is the percentage error in calculating b.

For analysis purposes, consider the minimum distance at which the pattern must be completely viewed to be

$$D_{\min} = \frac{2fR}{N_{b_{\max}} \Delta y} \tag{45}$$

where $N_{b_{\max}}$ is the number of pixels that would be calculated at such a distance. Let us then define the relative distance from the camera to the pattern as

$$m = \frac{D}{D_{\min}} = \frac{N_{b_{\max}}}{N_b}. \tag{46}$$

Then, substituting in (44),

$$|e| = 100 \frac{km}{N_{b_{\max}}}. \tag{47}$$

This indicates a linear relationship between m and $|e|$.

Equation (40) gives an expression for N_b; substituting into (44) results in

$$|e| = 100 \frac{kD\Delta y}{2fR}. \tag{48}$$

Again, this shows the linear relationship between the percentage error and the distance. In addition, the percentage error is also reduced if the parameter Δy is reduced, that is, if the resolution of the system is increased. Furthermore, this expression is inversely proportional to the focal length of the camera; this suggests that a closed-loop system, where the focal length is adjusted so that most of the pattern covers the image, will reduce the error in the system.

Let us now turn our attention to the effect that these errors will have on the calculation of the angle θ. This angle is calculated through the relationship shown in (10). Then, if N_a and N_b are the pixel counts for $2a$ and $2b$, respectively, and Δx and Δy are the lengths of the pixels in their respective dimensions,

$$\cos\theta = \frac{N_a \Delta x}{N_b \Delta y} = \alpha \left(\frac{N_a}{N_b} \right) \tag{49}$$

where α is the aspect ratio of the pixels. The pixel uncertainty will yield a range of calculation

$$\cos\bar{\theta} = \alpha \left(\frac{N_a \pm k}{N_b \mp k} \right) \tag{50}$$

where the effect of the pixel uncertainty will be worst at opposing signs, as implicit.

We have defined $N_{b_{\max}}$ as the pixel count for N_b at minimum distance. Defining $N_{a_{\max}}$ for N_a in the same way, we obtain

$$N_{a_{\max}} = \frac{2fR \cos\theta}{D_{\min} \Delta x}. \tag{51}$$

It is easily shown that the relative distance is also given by

$$m = \frac{N_{a_{\max}}}{N_a}. \tag{52}$$

Note that both maximum counts are related by

$$N_{a_{\max}} = \frac{N_{b_{\max}} \cos\theta}{\alpha}. \tag{53}$$

Replacing these results into (50),

$$\cos\bar{\theta} = \alpha \left(\frac{N_{a_{\max}} \pm km}{N_{b_{\max}} \mp km} \right). \tag{54}$$

Then, the absolute error in angle is given by

$$|\theta - \bar{\theta}| = \left| \theta - \arccos\left(\alpha \frac{N_{a_{\max}} \pm km}{N_{b_{\max}} \mp km} \right) \right|, \tag{55}$$

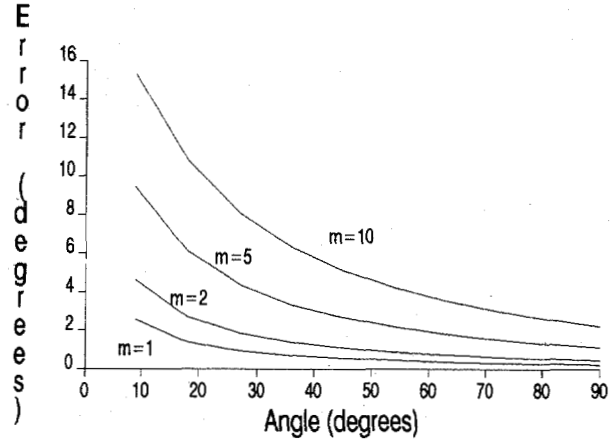
and the percentage error equals

$$|e|_\theta = \left| 1 - \frac{\arccos\left(\alpha \dfrac{N_{a_{\max}} \pm km}{N_{b_{\max}} \mp km} \right)}{\theta} \right| \times 100. \tag{56}$$

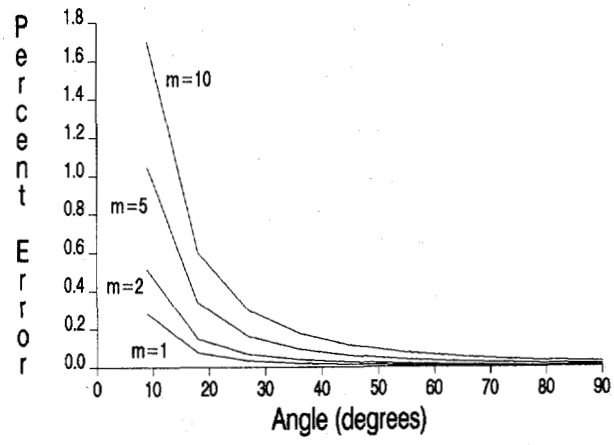
Both the absolute errors and relative errors have been plotted in Fig. 6 as a function of θ for various values of m and assuming $N_{b_{\max}} = 250$, $k = 1$, and $\alpha = 1$. We note that both the relative and absolute errors increase as θ becomes smaller; this is an obvious consequence of using the cosine of the angle to calculate it. This is perhaps one of the major drawbacks of the system since the calculations at the presumably ideal position are the most error prone. Note, however, that the errors also diminish at smaller distances, that is, when the ellipse is seen larger; then the errors can be made smaller by changing the focal length of the camera.

2) Swing Angle Errors: We have assumed that the swing σ of the camera is fixed such that a vertical line in the world will result in a vertical projection in the image plane. If this is not the case, the pattern will be seen as an ellipse tilted at an angle σ with respect to the vertical. Small variations in the swing are expected if the robot is in motion since irregularities on the floor, or improperly inflated tires, will generate such variations. It has already been noted that the swing angle can be appropriately accounted for by knowing that the bar codes are vertical; in addition to this, gravity sensors of some sort would also be quite helpful.

The general equation for an ellipse in x, y coordinates with



(a)



(b)

Fig. 6. Plots of error versus angle for various values of relative distance. (a) Absolute errors in degrees. (b) Percentage errors.

its center at the origin and tilted at an angle σ is

$$\left(\frac{\sin^2\sigma}{b^2}+\frac{\cos^2\sigma}{a^2}\right)x^2+\left(\frac{\sin^2\sigma}{a^2}+\frac{\cos^2\sigma}{b^2}\right)y^2 +2\sin\sigma\cos\sigma\left(\frac{1}{b^2}-\frac{1}{a^2}\right)xy=1. \quad (57)$$

The projections of this ellipse onto the $\hat{x}$ and $\hat{y}$ axes will be, respectively, the maximum values for x and y. In our case, these projections will constitute the calculated values for a and b in the presence of a swing σ on the camera; naming these calculated values a' and b', respectively, we obtain

$$a'^2=a^2\cos^2\sigma+b^2\sin^2\sigma \quad (58)$$

$$b'^2=b^2\cos^2\sigma+a^2\sin^2\sigma. \quad (59)$$

The calculated distance value will then be

$$D'=\frac{2fR}{2b'} \quad (60)$$

which can be shown to yield

$$D'=\frac{D}{\sqrt{\cos^2\sigma+\cos^2\theta\sin^2\sigma}}. \quad (61)$$

The percentage error can be calculated as

$$|e|_D=100\,\frac{|D-D'|}{D} =100\left(1-\frac{1}{\sqrt{\cos^2\sigma+\cos^2\theta\sin^2\sigma}}\right). \quad (62)$$

The most important observation to make is that the percentage error in distance due to the swing angle will not depend on the value of the distance itself; it does depend, however, on the value of the angle θ. As $\theta \to 0°$, $|e|_D \to 0$, which is natural since at $\theta = 0$ the pattern is seen as a circle, and the presence of nonideal swing will not make a difference. This property is quite important since it indicates that if θ can be properly adjusted so that the pattern is seen as a circle, an incorrect swing will not matter. As θ grows larger, it can be readily seen that the error will depend on the value of the swing angle σ, increasing until $\sigma = 90°$, and then decreasing until it reaches zero again at $\sigma = 180°$.

The calculated value for the cosine of the pan angle will be

$$\cos\theta'=\frac{a'}{b'} \quad (63)$$

which after some manipulation yields

$$\cos\theta'=\sqrt{\frac{\cos^2\theta+\sin^2\theta\sin^2\sigma}{1-\sin^2\theta\sin^2\sigma}}. \quad (64)$$

From here we can infer that the sine of the calculated pan angle will be

$$\sin\theta'=\sqrt{1-\cos^2\theta'} =\sqrt{\frac{\sin^2\theta\,(\cos^2\sigma-\sin^2\sigma)}{1-\sin^2\theta\sin^2\sigma}} \quad (65)$$

which is necessary since it is known that

$$\cos(\theta-\theta')=\cos\theta\cos\theta'-\sin\theta\sin\theta' =\cos\theta\sqrt{\frac{\cos^2\theta+\sin^2\theta\sin^2\sigma}{1-\sin^2\theta\sin^2\sigma}} +\sin^2\theta\sqrt{\frac{\cos^2\sigma-\sin^2\sigma}{1-\sin^2\theta\sin^2\sigma}}. \quad (66)$$

As $\theta \to 0°$, the error approaches zero, which again is only natural since at that point the pattern is seen as a circle and thus the swing will not affect it. If θ is not zero, the error is directly proportional to some function of σ and θ; notice, however, that there is a limitation on σ given in the second radical, that is

$$\cos^2\sigma-\sin^2\sigma\geq 0° \quad (67)$$

which yields the condition

$$-45°\leq\sigma\leq 45°. \quad (68)$$

Again, this is obvious if we note that any inclination of greater than 45° (positive or negative) will yield a calculated semiminor axis greater than the semimajor axis, and in this condition the cosine is not defined. It is not expected, however, that the swing angle vary so much.

Again, the error in θ does not depend on the distance from the pattern; it was noted in the distance calculation that if θ can be driven to zero, then the swing angle will not make a difference. If this is the objective, it is clear that this will present no problems; even if there are errors in measurements of θ due to the swing, these are only present if $\theta \neq 0°$. It is important to make the system such that the objective is to drive θ to 0°; in this case, a control of an adaptive nature would be appropriate since by correcting θ it will also be minimizing the error due to swing.

3) Sensor Height Errors: We have assumed that the height of the sensor is fixed so that it is exactly equivalent to the height of the center of the pattern. If a small variation exists in this height Δy, small errors will be introduced in the calculations. The reason for fixing the height at a certain value is to force β to $\pm 90°$; it is known that if β is not fixed at this value, then the ellipse will be tilted, making an angle β from the minor axis to the vertical. The effect of this tilting is the same as the effect made by the tilting due to the swing angle, with the ellipse tilted by $\beta - 90°$.

There is an additional effect, however, even if there is no tilting. Assume for now that $\beta = \pm 180°$; in this way the major and minor axes of the ellipse will lie on the coordinate axes. The first point to note is that if the pattern is still assumed centered, the vertical projection of the ellipse will be smaller than the horizontal projection, and thus the calculation of θ will not have a solution. (The real effect is that the major axis lies in the case at the horizontal; that is, the ellipse is tilted 90°, which as we saw before will not yield a solution.) This effect is quite disturbing; it can be solved, however, by forcing the optical axis of the camera to be always horizontal. Then assuming no lens distortion, which in any case is quite small, the projectional relationships still hold true.

Thus by forcing the optical axis to remain horizontal, the distance will be accurately calculated even in the pattern is not perfectly centered in the scene. The angle θ, however, will have a small error since it is being calculated with respect to a line perpendicular to the pattern but coplanar with both the center of the pattern and of the camera. (This is not coplanar with the optical axis since we have stated that the pattern is not centered anymore.) The error is obviously greatest at $\beta = 180°$; in this case the calculated angle θ' will be 0°, while the true value (with respect to the $\hat{z}$ axis) should be

$$\theta = \arctan \frac{\Delta y}{D}. \tag{69}$$

The error is obviously quite small for small Δy, and it is not even apparent why it would be an error at all. Rigorously, the angle should be taken with respect to the $\hat{z}$ axis, but the calculated angle will accurately determine the position at which θ is minimum, which ideally would be the working station.

B. General Case

Relaxing the assumptions we made before regarding the value of the co-azimuth angle β, we suggested the use of moments of the intrinsic parameters. We are currently working on an exhaustive analysis of the effect of various sources of error on the moment calculations themselves.

It is fairly safe to assume that since moment calculations are of an averaging nature, errors in each of the moments due to pixel uncertainties would probably be quite small; however, given the nature of (17)–(19), it is possible that the resulting error on the calculated values for D, θ, and β due to this factor could be at least of the same magnitude as the errors found for simple projection calculation, if not perhaps slightly more. We believe, however, that the added flexibility of the system greatly outweighs any loss of accuracy due to pixel uncertainties.

By using such a general method, there are no errors due to changes in height since the system will accurately find the position of the sensor in the spherical world coordinates. Consider first the case in which the camera has three degrees of freedom in translation motion; this could be achieved by being able to adjust the camera height or perhaps in a robot with flight capabilities. In this case, errors due to changes in height are irrelevant since the three-dimensional position of the sensor can always be adjusted.

If the translational motion of the camera, however, is only two dimensional, then the optimum three-dimensional position of the sensor would not be achievable in the presence of a change in height of the sensor. However, this change in height can be easily computed from the difference between the actual position and the desired position of the sensor. Thus an unexpected change in height can be appropriately posted to the user.

The effect of changes in swing angle will only be reflected in the calculation of the parameter β; no other parameter is affected. Supposing a swing angle σ with respect to the vertical, the computed co-azimuth angle β would be

$$\beta' = \beta + \sigma. \tag{70}$$

Note that changes on the swing angle can occur over a whole 360° domain without introducing unsolvable errors as was the case in the simplified case. The percentage error can be found to be

$$|e|_\beta = 100 \frac{\sigma}{\beta}. \tag{71}$$

On-line correction of this error would not be difficult since the angle σ can also be detected by finding the angle that a vertical line in the world (e.g., the bar codes) makes the vertical in the image.

Notice that by using the projections of the image, there were two instances in which the semiminor axis would be calculated greater than the semimajor axis. This instance would never be present if the calculations are done using moments.

The added flexibility of the moment equations determines that the system would not be liable to errors due to unexpected changes of the position of the sensor. Therefore, aside from

TABLE I
ERROR VALUES FOR EXPERIMENTAL COMPUTATIONS OF THE PAN ANGLE θ (ALL ANGLE VALUES ARE GIVEN IN DEGREES)

	Computed Angle			
	Greatest Error		Smallest Error	
Actual Angle	Absolute	Percent	Absolute	Percent
0	0.31	—	0.14	—
7.5	7.32	92.7	1.67	21.8
15	2.94	19.2	1.26	8.32
22.5	1.39	6.00	0.94	4.15
30	0.47	1.55	0.09	0.29
37.5	2.15	5.68	0.02	0.05
45	0.52	1.14	0.37	0.82
52.5	1.48	2.8	0.30	0.06
60	0.84	1.30	0.21	0.34
67.5	1.62	2.30	0.12	0.17
75	1.90	2.50	1.40	1.86

the error due to swing angle in β, the only errors that affect the system are those due to pixel uncertainties; such errors are indeed expected in any computer vision application.

VII. Experimental Results

Experimental results have been obtained for the simplest case in which the camera has been assumed at the same height from the pattern. A typical position calculation was performed by placing the camera over a gridwork of premeasured angles and distances; threshold values were determined by histogram methods to isolate the relative displacement pattern. From this thresholded image the length of the horizontal and vertical projections yield the length of the minor and major axes, respectively. The tilt angle θ and the distance D were obtained using the relationships specified before. Errors in calculated values with respect with premeasured angles and distances were computed and tabulated.

The height and swing angle of the camera were manually adjusted so that they would be at their correct positions; although small errors of this nature were expected, we believe that the errors would be mostly due to resolution and noise conditions. In addition, since the measured angles and distances were computed by hand, a certain amount of error due to human inaccuracy is also expected.

Table I shows the results obtained for the angle θ. Measurements were taken at several distances, and the maximum and minimum error in degrees were calibrated. From these values the percentage errors have also been computed. Notice that the percentage errors decrease as the angle grows larger, consistent with the deductions made before. The real errors do not follow such a steady pattern, but in general, they are also smaller for larger values of θ. As a useful comparison the minimum errors obtained also follow such a pattern.

The variations in the distance calculation were also tabulated, as shown in Table II. Measurements were done similarly, and the absolute and percentage errors were also tabulated. Notice that, as expected, the percentage error increases with distance. At a distance of 1 m, however, the error was relatively large; this is most probably due to inaccuracies in direct measurements.

TABLE II
PERCENTAGE ERROR FOR DISTANCE VALUES

Measured Distance (m)	Calculated Distance (m)	Percentage Error
1	0.98	2
2	2.01	0.50
3	2.99	0.33
4	4.07	1.78
5	5.11	2.20

VIII. Hardware Implementation

To perform the algorithms efficiently in real time, a dedicated hardware processor must be constructed; a block diagram of such a system is shown in Fig. 7. A color camera will be used to send data to a color coder circuit; this color coder will be designed to differentiate between the various regions of the pattern. Initially, the start/stop bar codes will be scanned for; if these are not found, the circuit will signal the camera mount control to change the angle of view.

Once the bar codes are found, the system knows it is viewing the pattern; moment generation is therefore enabled. This section will calculate the moments for the image and shadow patterns in real time. This calculation will be carried out using only counters, shift registers, and accumulators, as has been seen previously. Once the whole image has been processed, an interrupt signal is generated to inform the CPU that data are available.

The hardware will be supported by two software programs; the first functions as a hardware driver, and the second one scans for the pattern and processes the calculated moments to verify the robot position, and, if necessary, generate appropriate control signals to the camera servos.

We are currently working on the development of this system. Software simulation results have been encouraging.

IX. Conclusion

A simple and inexpensive method for position verification for a mobile robot using standard patterns has been developed based on the geometric properties of the projection of a circle in the image plane of the camera. This projection always results in an ellipse whose size, eccentricity, and orientation have been used to completely determine the distance, elevation, and co-azimuth angles D, θ, and β of the viewing point with respect to the center of the circle, with a remaining ambiguity in β. This ambiguity was solved by a simple modification of the pattern. By computing the relative position of the camera with respect to the circle in three dimensions, the pattern can be placed at any position in the environment, adding great flexibility to the procedure of environment layout.

The pattern was further enhanced by associating a unique code to it such that an environment in which multiple working stations are necessary can be appropriately handled. These

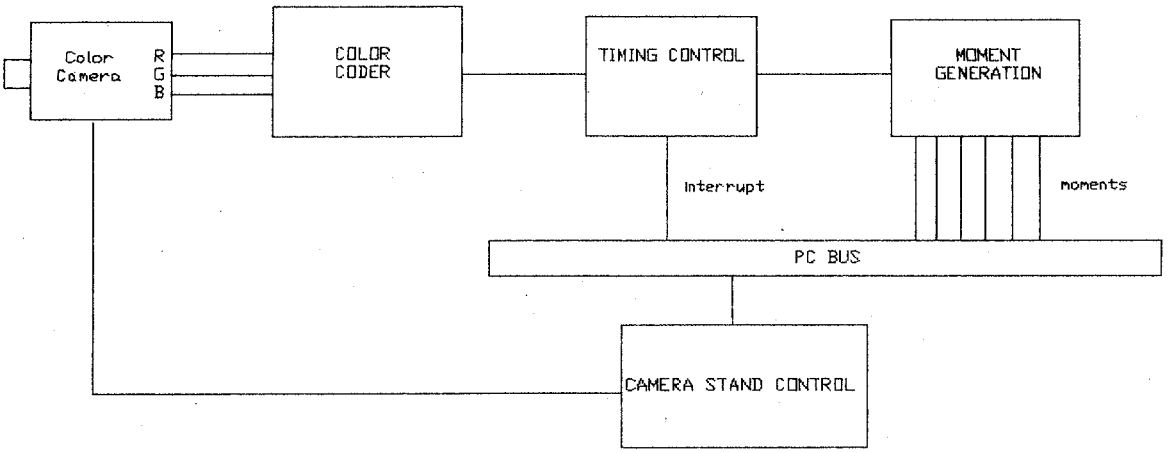


Fig. 7. Hardware system block diagram.

codes were also used to simplify the procedure of scanning for the patterns in the environment, yielding an overall solution that is readily implementable in industry.

A thorough analysis of several sources of error that can adversely affect the performance of the system has been done for the restricted case in which $\beta = \pm 90°$; this analysis was compared against experimental results that were obtained. This analysis has shown that while there are several sources of error that affect the algorithms, most of their effects are quite small. Some qualitative observations of the possible effects of the same sources of error in the general case where β is unrestricted have also been presented. These observations indicate that the general method is indeed more suitable for implementation.

We are currently working on developing a thorough analysis of the moment algorithms toward the calculation of the relative displacement. The results of this analysis will be used to develop a special-purpose hardware system that would handle this task in real time.

Acknowledgment

The authors would like to thank P. Shironoshita and B. Hussain for their contributions. The authors also wish to thank the referees for their comments and suggestions.

References

[1] M. H. E. Larcommbe, "Tracking stability of wire guided vehicles," in *Proc. Int. Conf. Auto. Guided Veh. Syst.*, June 1981, pp. 137–144.

[2] Keith C. Drake, E. S. McVey, and R. M. Iñigo, "Sensing error for a mobile robot using line navigation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-7, no. 4, pp. 485–490, July 1985.

[3] E. S. McVey, K. C. Drake, and R. M. Iñigo, "Range measurements by a mobile robot using a navigation line," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-8, pp. 105–109, Jan. 1986.

[4] M. Julliere, L. Marce, and H. Place, "A guidance system for a mobile robot," in *Proc. 13th Int. Symp. Indust. Robots Robot. 7,* 1983, vol. 2, pp. 58–68.

[5] R. A. Cooke, "Microcomputer control of free ranging robots," in *Proc. 13th Int. Symp. Indust. Robots Robot. 7,* 1984, vol. 2, pp. 109–120.

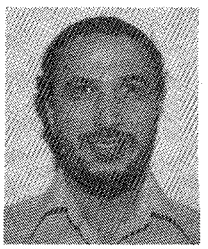
[6] H. P. Moravec, "Visual mapping by a robot rover," in *Proc. 6th Joint Int. Conf. Artif. Intell.*, Aug. 1979, pp. 589–600.

[7] I. Fukui, "TV image processing to determine the position of a robot vehicle," *Pattern Recognition,* vol. 14, nos. 1–6, pp. 101–109, 1981.

[8] M. J. Magee and J. K. Aggarwal, "Determining the position of a robot using a single calibration object," in *Proc. Int. Conf. Robotics,* Mar. 1984, pp. 140–150.

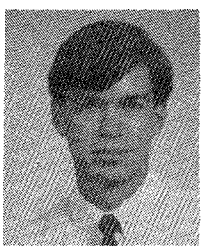
[9] *Microprocessor Based Bar-Code Design,* Hewlett Packard, Appl. Note 1014, Nov. 1982.

[10] R. L. Andersson, "Real-time gray-scale video processing using a moment-generating chip," *IEEE J. Robotics Automat.*, vol. RA-2, pp. 79–85, June 1985.



Mansur R. Kabuka (S'77–M'78–S'81–M'82) received the B.S. degree in electrical engineering and computer science from the University of Alexandria, the M.S. degree from the University of Miami, Coral Gables, FL, and the Ph.D. degree from the University of Virginia, Charlottesville.

In 1983, he joined the Faculty of the Department of Electrical and Computer Engineering of the University of Miami where he is currently an Assistant Professor. During the summer of 1984, he worked for IBM Corporation, Boca Raton, FL. His research interests include computer vision and robotics.



Alvaro E. Arenas was born in Cali, Colombia in 1957. He received the B.Sc. degree in mechanical engineering from the University of Miami, Coral Gables, FL, in 1980. He is currently working toward the M.Sc. degree in electrical engineering.

He is currently employed at the Cordis Corporation as a Product/Process Development Engineer and is involved with the introduction of a percutaneous transluminal coronary (PTCA) dilitation catheter.