Pattern Recognition, Vol. 19, No. 6, pp. 439–451, 1986.
Printed in Great Britain.

0031-3203/86 $3.00 + .00
Pergamon Journals Ltd.
Pattern Recognition Society

A NEW APPROACH TO ROBOT LOCATION BY HOUSE CORNERS

HUEI-LIN CHOU
Institute of Computer Engineering, National Chiao Tung University, Hsinchu, Taiwan 300, Republic of China

and

WEN-HSIANG TSAI*
Department of Information Science, National Chiao Tung University, Hsinchu, Taiwan 300, Republic of China

(*Received* 10 *September* 1985; *received for publication* 27 *January* 1986)

Abstract—A new approach to robot location in an in-house 3-D space using house corners as the standard mark is proposed. A monocular image of a house corner is first taken. Image processing and numerical analysis techniques are then applied to find the equations of the three lines going through the corner point. Under the reasonable assumption that the distance from the camera to the ceiling is known in advance, the position of the robot, on which the camera is mounted, is finally uniquely determined according to 3-D imaging geometry. Experimental results with location error less than 5% on the average prove the feasibility of the proposed approach. Error analysis useful for determining location precision is also included.

Robot location
Position parameters
Image analysis
Direction parameters
Imaging geometry
Camera calibration

1. INTRODUCTION

With the advent of mobile robots,[1–2] it is important to be able to locate a robot while it is navigating around a controllable environment. One way for robot location is to use image analysis techniques,[2–6] provided that the robot is equipped with visual sensors like TV cameras. To simulate human stereo vision, binocular images usually are taken and the scene correspondence problem is solved for 3-D range finding.[2] However, within a controllable environment like a warehouse, monocular images are found sufficient for robot location if special "standard marks" are adopted for image analysis.[3–6]

Fukui[3] placed on the wall as the standard mark a diamond shape whose boundary consists of four identical thick line segments with a known length (see Fig. 1). The two diagonals of the diamond are arranged to be vertical and horizontal, respectively. The lens center of a TV camera is put at the same height as the diamond center, and images are taken when the camera optical axis is pointed through the diamond center. The boundary of the diamond image is then extracted, the lengths of the two diagonals computed, and the location of the camera finally derived, based on the two diagonals in the image.

Courtney and Aggarwal[4–5] used the same standard mark as Fukui's but relaxed the restriction that the camera lens center must be as high as the diamond center. Instead, they made the reasonable assumption that the height of the camera is known.

Magee and Aggarwal[6] used a sphere as the standard mark on which two perpendicular great circles are drawn for use in robot location (see Fig. 2). Again, before the image of the sphere is taken, the camera optical axis must be pointed through the sphere center. The distance from the camera to the sphere center is computed from the size of the projected circle of the sphere, and the direction of the camera is computed from the points on the projections of the great circles that are closest to the center of the sphere's outline. The camera location is finally computed accordingly in terms of sphere coordinates.

In all the previous approaches, artificially-created landmarks must be used, and the camera optical axis must be pointed through the mark center. In this study, we utilize house corners as the standard mark. Such marks exist in most houses, usually are visible from house grounds, and most are of an identical geometric structure, a "Y" shape (see Fig. 3 for an example). The height of each corner which, in most cases, is part of the house ceiling can also be reasonably assumed to be known in advance.

A Y-shape corner, which we use for robot location in this study, is composed of three perpendicular planes, or alternatively, of three perpendicular lines with each

* Author to whom correspondence should be sent.

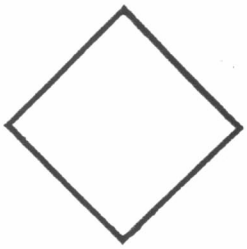


Fig. 1. A diamond-shaped standard mark.

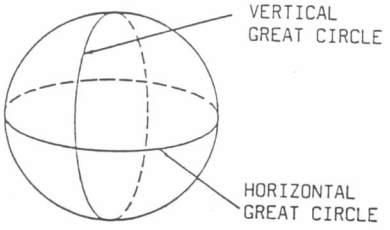


Fig. 2. A sphere used for robot location.

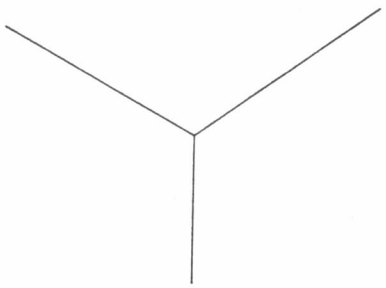


Fig. 3. The perspective projection of a house corner.

line being the intersection of a pair of planes. The three lines intersect at the corner point to form the shape Y. It can be easily observed that the Y shape of the corner varies relatively with respect to the viewer's location and angle. And this is the basic principle behind the proposed approach.

More specifically, in the approach a monocular image of a house corner with the Y shape is taken whenever the robot is to be located. The projections of the three lines going through the corner on the image plane are then extracted and fitted in the least square error sense by three line equations. Under the reasonable assumption that the heights of the camera and the corner are known (or simply that the distance from the camera to the ceiling is known), the coefficients of the equations finally are substituted into a set of location formula to get the camera (i.e., the robot) position as well as direction.

In the remainder of this paper, the location formula are derived in Section 2 according to 3-D imaging geometry. The image processing and numerical techniques used for corner extraction and line fitting are described in Section 3. To improve location precision, image distortion correction and system parameter calibration are also included, as described in Section 4. Experimental results and error analysis are included in Section 5. Conclusions and some suggestions for further research are found in Section 6.

2. DERIVATION OF ROBOT LOCATION BY CORNER IMAGE

Giving a monocular image of a Y-shaped house corner, let the equations of the three lines through the corner point in terms of image coordinates (u_p, v_p) be described by $u_p + b_i v_p + c_i = 0$, $i = 1, 2, 3$. The derivation of these equations will be described in Section 3. The robot or the camera location will be described with respect to a global coordinate system formed by the corner structure. In the global system, the corner point is used as the origin, the two perpendicular lines on the ceiling as the X and Y axes, and the third line through the corner as the Z axis. In addition, a camera coordinate system with U, V and W axes is also established, in which the origin is located at the camera lens center, the W axis is aligned with the camera optical axis, the $U-V$ plane is parallel to the image plane, and the U axis is parallel to the $X-Y$ plane of the global system. The image plane is located at $W = f$, where f is the focus length of the camera. The geometrical relation between the global and the camera coordinate system is illustrated in Fig. 4.

The desired camera location will be described by three position parameters X_c, Y_c and Z_c and three direction parameters ψ, Θ and δ, where Z_c is the distance from the camera to the ceiling (i.e. the $X-Y$ plane) assumed to be known, and ψ, Θ and δ are called the pan, tilt and swing angles of the camera, respectively, with respect to the global coordinate system. In the sequel, these six camera location parameters will be derived in terms of the six coefficients $b_1, c_1, b_2, c_2, b_3, c_3$ of the equations of the three lines in the corner image. The coordinate transformation from the global coordinate system to the camera system will first be described in terms of the homogeneous coordinates.[7–9] The six coefficients b_i, c_i, $i = 1, 2, 3$, of the three line equations will then be derived in terms of the six camera parameters. The case where the image plane is not rotated (i.e. the swing angle $\delta = 0$) is first considered. It is then extended to the more general case with the image plane being rotated. The result will consist of six equalities which finally can be solved to obtain the desired camera location parameters.

2.1. *Coordinate transformation from camera system to global system*

It is more convenient to describe imaging geometry in terms of the camera coordinates, so we first

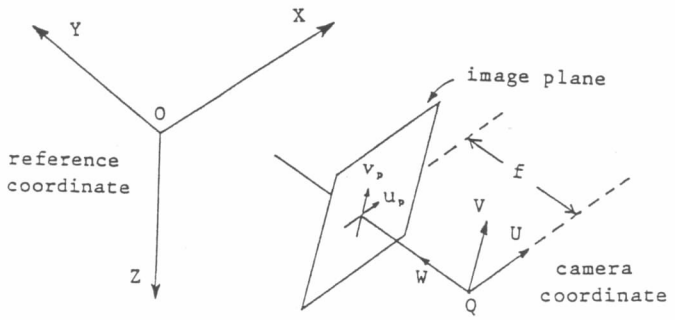


Fig. 4. The camera coordinate system and the global coordinate system.

transform the global coordinates into the camera coordinates. This can be accomplished by one origin translation, two axis rotations, and one axis reversion for the case $\delta = 0$ as follows. Extension to the case $\delta \neq 0$ will be discussed later.

(1) Translate the origin of the global system by $(-X_c, -Y_c, -Z_c)$ to the origin of the camera system.

(2) Rotate the $X - Y$ plane about the Z axis through the pan angle ψ such that the X axis is parallel to the U axis.

(3) Rotate the Y–Z plane about the X axis through the tilt angle Θ such that the X–Y plane is parallel to the U–V plane.

(4) Reverse the Z axis such that the positive direction of the Z axis is identical to the negative direction of the W axis. This is necessary because the camera system is a left-handed coordinate system while the global system is a right-handed one.

Let P be any point in the 3-D space with global coordinates (x, y, z) and camera coordinates (u, v, w), then the above coordinate transformation can be described as follows[7–9]:

$$(u, v, w, 1) = (x, y, z, 1) \cdot T(X_c, Y_c, Z_c) \cdot R_z(\psi) \cdot R_x(\Theta) \cdot F_z = (x, y, z, 1)T_r \tag{1}$$

where

$$T(X_c, Y_c, Z_c) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -X_c & -Y_c & -Z_c & 1 \end{bmatrix},$$

$$R_z(\psi) = \begin{bmatrix} \cos\psi & -\sin\psi & 0 & 0 \\ \sin\psi & \cos\psi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$R_x(\Theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\Theta & -\sin\Theta & 0 \\ 0 & \sin\Theta & \cos\Theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$F_z = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

and

$$T_r = T(X_c, Y_c, Z_c)R_z(\psi)R_x(\Theta)F_z = \begin{bmatrix} \cos\psi & -\sin\psi\cos\Theta & -\sin\psi\sin\Theta & 0 \\ \sin\psi & \cos\psi\cos\Theta & \cos\psi\sin\Theta & 0 \\ 0 & \sin\Theta & -\cos\Theta & 0 \\ x_0 & y_0 & z_0 & 1 \end{bmatrix}$$

with

$$\begin{aligned} x_0 &= -X_c\cos\psi - Y_c\sin\psi, \\ y_0 &= (X_c\sin\psi - Y_c\cos\psi)\cos\Theta - Z_c\sin\Theta, \\ z_0 &= (X_c\sin\psi - Y_c\cos\psi)\sin\Theta + Z_c\cos\Theta. \end{aligned} \tag{2}$$

2.2. *Derivation of equation coefficients of three corner lines*

The coefficients b_i and c_i of the line equation $u_p + b_i v_p + c_i = 0$ for the projection of the X axis in the image plane will first be derived for the case of zero swing angle (i.e. $\delta = 0$). Let P be any point on the X axis in the 3-D space with global coordinates $(x, 0, 0)$ and camera coordinates (u_x, v_x, w_x). Let (u_p, v_p) be the image coordinates of the projection of P on the image plane. Then, according to imaging geometry,[10] we have

$$\begin{aligned} u_p &= fu_x/w_x, \\ v_p &= fv_x/w_x, \end{aligned} \tag{3}$$

where f is the camera focus length. In addition, from equation (1), we have

$$\begin{aligned} (u_x, v_x, w_x, 1) &= (x, 0, 0, 1) \cdot T_r \\ &= (x\cos\psi + x_0, -x\sin\psi\cos\Theta + y_0, \\ &\quad -x\sin\psi\sin\Theta + z_0, 1). \end{aligned}$$

Substituting the values of u and v above into the two equalities in equation (3), we get

$$u_p = \frac{f(x\cos\psi + x_0)}{-x\sin\psi\sin\Theta + z_0},$$

$$v_p = \frac{f(-x\sin\psi\cos\Theta + y_0)}{-x\sin\psi\sin\Theta + z_0}.$$

Eliminating the variable x in the above two equalities, we get the following equation for the X axis in the image plane:

$$u_p = \frac{v_p(-z_0\cos\psi - x_0\sin\psi\sin\Theta) + f(y_0\cos\psi + x_0\sin\psi\cos\Theta)}{-y_0\sin\psi\sin\Theta + z_0\sin\psi\cos\Theta}.$$

which, when expressed in the form $u_p + b'_1 v_p + c'_1 = 0$, leads to the following two equalities after x_0, y_0 and z_0 are substituted by the values in (2):

$$b'_1 = \frac{Y_c\sin\Theta - Z_c\cos\psi\cos\Theta}{-Z_c\sin\psi},$$

$$c'_1 = \frac{f(-Y_c\cos\Theta - Z_c\cos\psi\sin\Theta)}{-Z_c\sin\psi}. \quad (4)$$

When the camera swing angle δ is nonzero (i.e. when the image plane is rotated through angle $\delta \neq 0$), let (u_i, v_i) be the rotated version of (u_p, v_p), then

$$(u_p, v_p) = (u_i, v_i)\begin{bmatrix}\cos\delta & -\sin\delta\\ \sin\delta & \cos\delta\end{bmatrix}$$

$$= (u_i\cos\delta + v_i\sin\delta,\ -u_i\sin\delta + v_i\cos\delta).$$

And the equation $u_p + b'_1 v_p + c'_1 = 0$ for the X axis in the image plane now becomes

$$(u_i\cos\delta + v_i\sin\delta) + b'_1(-u_i\sin\delta + v_i\cos\delta) + c'_1 = 0$$

or equivalently

$$u_i + \frac{\sin\delta + b'_1\cos\delta}{\cos\delta - b'_1\sin\delta}v_i + \frac{c'_1}{\cos\delta - b'_1\sin\delta} = 0$$

which, when expressed in the form $u_i + b_1 v_i + c_1 = 0$, leads to the following two desired equalities after b'_1 and c'_1 are substituted by the values in (4):

$$b_1 = \frac{-Z_c\sin\psi\sin\delta + (Y_c\sin\Theta - Z_c\cos\psi\cos\Theta)\cos\delta}{-Z_c\sin\psi\cos\delta - (Y_c\sin\Theta - Z_c\cos\psi\cos\Theta)\sin\delta}, \quad (5)$$

$$c_1 = \frac{f(-Y_c\cos\Theta - Z_c\cos\psi\sin\Theta)}{-Z_c\sin\psi\cos\delta - (Y_c\sin\Theta - Z_c\cos\psi\cos\Theta)\sin\delta}. \quad (6)$$

Similarly, the coefficients of the line equation $u_i + b_2 v_i + c_2 = 0$ for the projection of the Y axis in the image plane can be derived to be

$$b_2 = \frac{Z_c\cos\psi\sin\delta + (-X_c\sin\Theta - Z_c\sin\psi\cos\Theta)\cos\delta}{Z_c\cos\psi\cos\delta - (-X_c\sin\Theta - Z_c\sin\psi\cos\Theta)\sin\delta}, \quad (7)$$

$$c_2 = \frac{f(X_c\cos\Theta - Z_c\sin\psi\sin\Theta)}{Z_c\cos\psi\cos\delta - (-X_c\sin\Theta - Z_c\sin\psi\cos\Theta)\sin\delta}, \quad (8)$$

and those for the Z axis be

$$b_3 = \frac{(X_c\sin\psi - Y_c\cos\psi)\sin\delta + (X_c\cos\psi + Y_c\sin\psi)\cos\Theta\cos\delta}{(X_c\sin\psi - Y_c\cos\psi)\cos\delta - (X_c\cos\psi + Y_c\sin\psi)\cos\Theta\sin\delta}, \quad (9)$$

$$c_3 = \frac{f(X_c\cos\psi + Y_c\sin\psi)\sin\Theta}{(X_c\sin\psi - Y_c\cos\psi)\cos\delta - (X_c\cos\psi + Y_c\sin\psi)\cos\Theta\sin\delta}. \quad (10)$$

The details of derivation are included in Ref. (11) and are omitted here.

2.3. *Computation of camera location parameters*

Among the six equalities (5) through (10) derived above, only five are independent because the projections of the three axes all go through a common point, the corner, in the image plane. The values of the equation coefficients b_i and c_i, $i = 1, 2, 3$, will be derived in Section 3 using image processing and numerical analysis techniques. The value Z_c is assumed to be known already, and the focus length f can be found by calibration as described in the next section. Therefore, the remaining five variables X_c, Y_c, ψ, Θ and δ can be solved with the five independent equalities in (5) through (10). The following is the procedure to get the solution.

First, equation (5) can be transformed into

$$Z_c(-b_1\sin\psi\cos\delta + b_1\cos\psi\cos\Theta\sin\delta + \sin\psi\sin\delta + \cos\psi\cos\Theta\cos\delta) = Y_c(\sin\Theta\cos\delta + b_1\sin\Theta\sin\delta).$$

And equation (6) can be transformed into

$$Z_c(-c_1\sin\psi\cos\delta + c_1\cos\psi\cos\Theta\sin\delta + f\cos\psi\sin\Theta) = Y_c(-f\cos\Theta + c_1\sin\Theta\sin\delta).$$

Eliminating Z_c and Y_c from the above two equalities, we get

$$fb_1 \sin\psi \cos\Theta \cos\delta - fb_1 \cos\psi \sin\delta - f \sin\psi \cos\Theta \sin\delta - f \cos\psi \cos\delta + c_1 \sin\psi \sin\Theta = 0$$

which, after divided by $\cos\psi$, can be reduced to

$$\tan\psi = \frac{fb_1 \sin\delta + f\cos\delta}{fb_1 \cos\Theta \cos\delta - f\cos\Theta \sin\delta + c_1 \sin\Theta}. \tag{11}$$

Applying similar steps to equations (7) and (8), we can get

$$\tan\psi = \frac{-(fb_2 \cos\Theta \cos\delta - f\cos\Theta \sin\delta + c_2 \sin\Theta)}{fb_2 \sin\delta + f\cos\delta}. \tag{12}$$

Equating (11) and (12) above and considering $\tan\Theta$ as a variable, we can transform the result into the following equation

$$A_\Theta \tan^2\Theta + B_\Theta \tan\Theta + C_\Theta = 0 \tag{13}$$

where

$$A_\Theta = f^2(b_1 \sin\delta + \cos\delta)(b_2 \sin\delta + \cos\delta) + c_1c_2,$$

$$B_\Theta = f(c_1(b_2 \cos\delta - \sin\delta) + c_2(b_1 \cos\delta - \sin\delta)),$$

$$C_\Theta = f^2(1 + b_1b_2).$$

On the other hand, equations (9) and (10) can be reduced to the following two equations, respectively:

$$X_c(b_3 \sin\psi \cos\delta - b_3 \cos\psi \cos\Theta \sin\delta - \sin\psi \sin\delta - \cos\psi \cos\Theta \cos\delta) = Y_c(-\cos\psi \sin\delta + \sin\psi \cos\Theta \cos\delta + b_3 \cos\psi \cos\delta + b_3 \sin\psi \cos\Theta \sin\delta),$$

$$X_c(c_3 \sin\psi \cos\delta - c_3 \cos\psi \cos\Theta \sin\delta - f\cos\psi \sin\Theta) = Y_c(f \sin\psi \sin\Theta + c_3 \cos\psi \cos\delta + c_3 \sin\psi \cos\Theta \sin\delta)$$

which, after X_c and Y_c are eliminated, can be combined and reduced simply to

$$fb_3 \sin\Theta \cos\delta - f\sin\Theta \sin\delta - c_3 \cos\Theta = 0,$$

or equivalently, to

$$\tan\Theta = \frac{c_3}{f(b_3 \cos\delta - \sin\delta)}. \tag{14}$$

Note that no term of $\sin\psi$ or $\cos\psi$ is involved in the above equality; they are eliminated in the reduction. Now, substituting (14) (i.e. the value of $\tan\Theta$ in terms of $\sin\delta$ and $\cos\delta$) into (13) and reducing the resultant equation into a form with $\tan\delta$ as a variable, we get

$$A_\delta \tan^2\delta + B_\delta \tan\delta + C_\delta = 0 \tag{15}$$

where

$$A_\delta = f^2c_3^2b_1b_2 + c_3^2c_1c_2 + f^2c_1c_3 + f^2c_2c_3 + f^4(1 + b_1b_2),$$

$$B_\delta = f^2c_3^2(b_1 + b_2) + f^2c_3c_1(b_2 + b_3) - f^2c_2c_3(b_3 + b_1) - 2f^4b_3(1 + b_1b_2),$$

$$C_\delta = f^2c_3^2 + c_3^2c_1c_2 + f^2c_1c_3b_2b_3 + f^2c_2c_3b_1b_3 + f^4(1 + b_1b_2)b_3^2.$$

Now, the coefficients A_δ, B_δ and C_δ are all expressed in terms of known values b_i, c_i and f. Therefore, equation (15) can be solved to obtain $\tan\delta$ as

$$\tan\delta = \frac{-B_\delta \pm \sqrt{B_\delta^2 - 4A_\delta C_\delta}}{2A_\delta} \tag{16}$$

where the solution with a minus sign before the square root is found by experiments to be the desired one. After the valve δ is computed, it can be substituted into equation (14) to obtain the value $\tan\Theta$. Thereafter, the values of δ and Θ can be substituted into equations (11) or (12) to obtain the value $\tan\psi$. Finally, the value of δ, Θ, ψ and Z_c can be substituted into equations (5) and (7) to obtain the values of Y_c and X_c, respectively. This completes the derivation of the camera location parameters X_c, Y_c, ψ, Θ and δ.

3. FINDING CORNER LINES BY IMAGE ANALYSIS TECHNIQUES

The purpose of image and numerical analysis in this study is to find the three lines going through the corner in each corner image and then to compute the coefficients of the line equations which are used as input data for camera location computation as described in last section. The essence here is that the equation coefficients should be computed as accurate as possible so that the location of the camera can be computed more precisely. This is fulfilled by applying the least-square-error (LSE) fitting twice to the image points of the three corner lines found by a sequence of image processing operations. The procedure of corner line finding is shown in Fig. 5 whose details are discussed next.

3.1. *Edge detection and thinning*

The three lines going through the corner can be found by edge detection and thinning, if the lighting on each wall is different from that on another. An example of corner images is shown in Fig. 6, in which different lighting on the walls result in different gray value ranges in the images. Sobel edge operators are used to compute the edge magnitudes for the horizontal and vertical directions, respectively.[12] Pixels with edge magnitudes larger than a threshold value are detected as edge points. The result of edge detection in the horizontal direction is shown in Fig. 7(a), and that in the vertical direction is shown in Fig. 7(b).

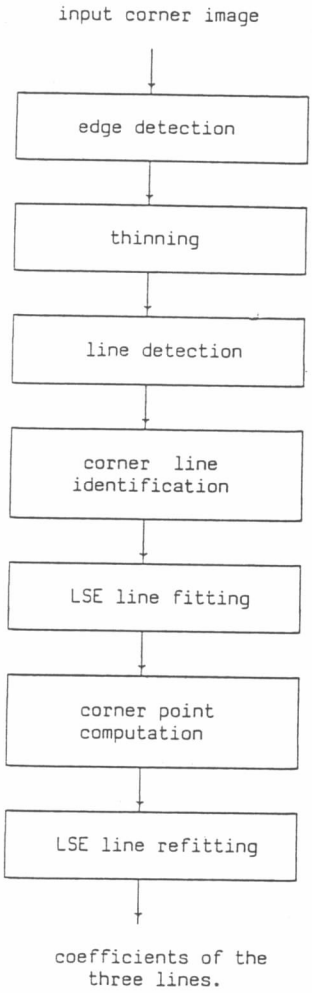


Fig. 5. The block diagram of corner line finding.

The fast thinning algorithm proposed by Shanmugan and Paul[13] is then adopted to thin the edge points found in the horizontal direction and in the vertical direction, respectively. The results in the two directions are superimposed next to get the final thinning result. The result of applying thinning to Figs. 7(a) and (b) is shown in Fig. 8.

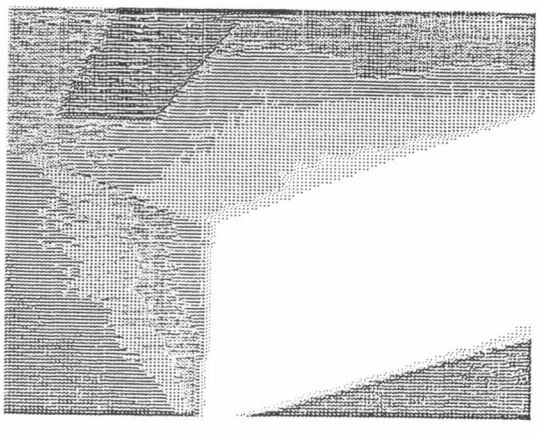


Fig. 6. A corner image.

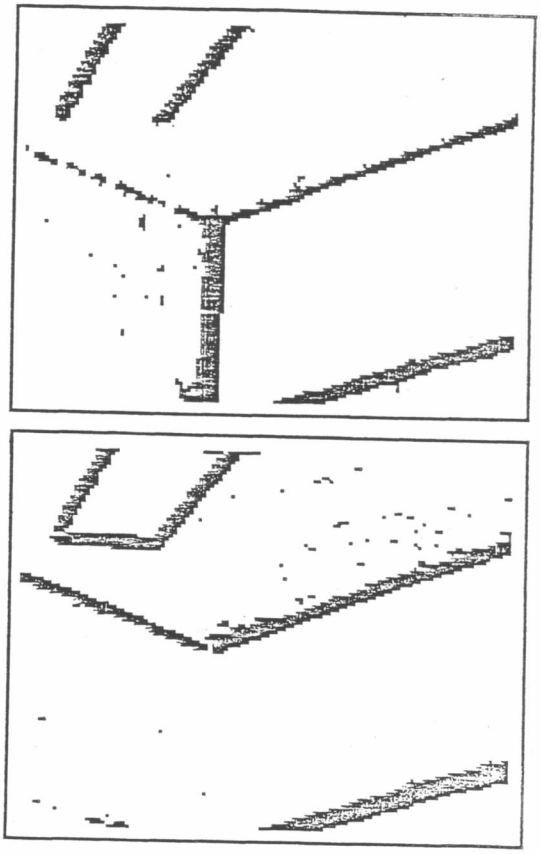


Fig. 7. (a) Edge points in the horizontal direction, (b) edge points in the vertical direction.

3.2. *Hough transform for line detection*

The result of thinning is a set of edge points which might include some noise points not necessarily lying on the boundary of two walls. To detect the boundary lines between the walls, the Hough transform[14] is applied. An advantage of using the Hough transform for line detection is that it can endure noise and gaps to some limit.

In the Hough transform we apply in this study, each line is represented by the following parametric form

$$x \cos \alpha + y \sin \alpha = \rho$$

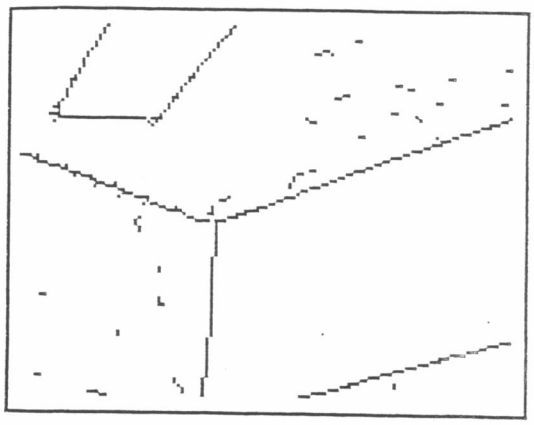


Fig. 8. The result of thinning.

where x and y are the image coordinates of the line points, α is the direction of the normal of the line, and ρ is the distance of the line to the origin of the image coordinate system. Each point in the image coordinate space is transformed into a curve in the parametric space with α and ρ as the coordinates. All the curves corresponding to a set of image points lying on a straight line will intersect at a point in the parametric space. A so-called cell array $H(\alpha, \rho)$ is created to count the number of curve points going through each point (α, ρ) in the parametric space. If the cell value $H(\alpha_0, \rho_0)$ at (α, ρ_0) is found to be locally maximum and larger than a predetermined threshold, then it is decided that there exists a corresponding line in the image coordinate space with the equation

$$x \cos \alpha_0 + y \sin \alpha_0 = \rho_0.$$

In the real case where noise causes the boundary points to fall about, instead of exactly on, a straight line, the corresponding curves in the parametric space will not intersect right at a single point (α_0, ρ_0), but around the desired point (α_0, ρ_0). To take care of this, each $H(\alpha, \rho)$ value is replaced by $H'(\alpha, \rho) = H(\alpha, \rho) + H(\alpha, \rho + 1) + H(\alpha, \rho - 1)$. Figure 9 includes the cell array $H'(\alpha, \rho)$ resulting from applying the Hough transform to Fig. 8. Only cell values larger than 15 are shown. All locally maximum values larger than 40 are squared. Using these (α_0, ρ_0) values, the points of the corresponding lines in the image plane can be found, as shown in Fig. 10.

3.2. *Identification of corner lines*

Any line in the image with its length long enough will be detected by the Hough transform process described above. There might exist more than three lines in the result. It remains to identify among the detected lines the three which consist of the Y-shaped corner. This is accomplished in the proposed approach by the following condition checks:

(1) there must exist a triple of lines each of which has an end point very close to that of another (note that in the real case, the three lines need not go through a common point owing to the noise or distortion caused by image taking and processing);

(2) the intersection point of every pair of lines in the triple must be very close to that of another pair (in the real case, the three intersection points need not be located at a single common point);

(3) each line in the triple must have an end point [other than the one mentioned in (1)] which is close to the image boundary;

(4) the angle between every pair of lines in the triple must have a value larger than 90° and smaller than 180°.

The result of corner line identification applied to Fig. 10 according to the above checks is shown in Fig. 11.

3.4. *LSE fitting for corner line approximation*

The normal parameters of the line equations found by the Hough transform is not accurate enough for the purpose of robot location. Improvement on this is made by fitting the point set of each line found previously (see Fig. 11) to a line equation in the least-square-error sence. Let the line points to be fitted be denoted by $S = \{(x_i, y_i) | i = 1, 2, \ldots, n\}$, and let the fitting line L be described by $\rho = x \cos \alpha + y \sin \alpha$. Then, LSE line fitting[15] requires that the values ρ and α be such that the square error (computed as the sum of the squares of all the distances from the points in S to L)

$$E^2 = \sum_{i=1}^{n} (\rho - x_i \cos \alpha - y_i \sin \alpha)^2 \qquad (17)$$

is minimized for all ρ and α. Differentiating E^2 with respect to ρ and α, setting the results to be zero, and solving the resulting two equations, we can obtain the desired line parameters as

$$\alpha = \tan^{-1} [(A - \sqrt{A^2 + 4B^2})/2B],$$

$$\rho = (\cos \alpha \Sigma x + \sin \alpha \Sigma y)/n$$

where Σ means $\sum_{i=1}^{n}$ and

$$A = \Sigma y^2 - \Sigma x^2 + [(\Sigma x)^2 - (\Sigma y)^2]/n,$$

$$B = \Sigma xy - (\Sigma x \Sigma y)/n.$$

3.5. *LSE refitting for more accurate corner line approximation*

The lines found by LSE fitting in the last step need not all go through a common point which is supposed to be the corner point. Again, this is due to the noise or distortion caused by image digitization and processing. Further improvement on corner approximation accuracy can be obtained by the following three steps:

(1) use the three line equations found by previous LSE fitting to compute three intersection points, each being that of a line pair;

(2) find the geometric center of the three intersection points, which is then regarded as the desired corner point C;

(3) refit the point set of each line found by the Hough transform to a new line equation under the constraint that the point C computed above should be passed.

Let C be located at the coordinates (x_0, y_0). Then, the third step above defines a new constrained LSE line fitting problem, namely, giving a point set $S = \{(x_i, y_i) | i = 1, 2, \ldots, n\}$, try to find a line L with the equation $\rho = x \cos \alpha + y \sin \alpha$ such that the square error E^2 described by (17) is minimized for all ρ and α, and the equality $\rho - x_0 \cos \alpha + y_0 \sin \alpha = 0$ is satisfied (i.e. the condition that L passes C is satisfied). This problem can be solved by first translating the origin of the (x, y) coordinate system to the new origin (x_0, y_0) to form a new (x', y') coordinate system, so that each point (x_i, y_i) in S now becomes $(x_i', y_i') = (x_i - x_0, y_i - y_0)$, the constraint $\rho - x_0 \cos \alpha - y_0 \sin \alpha = 0$ becomes $\rho = 0$,

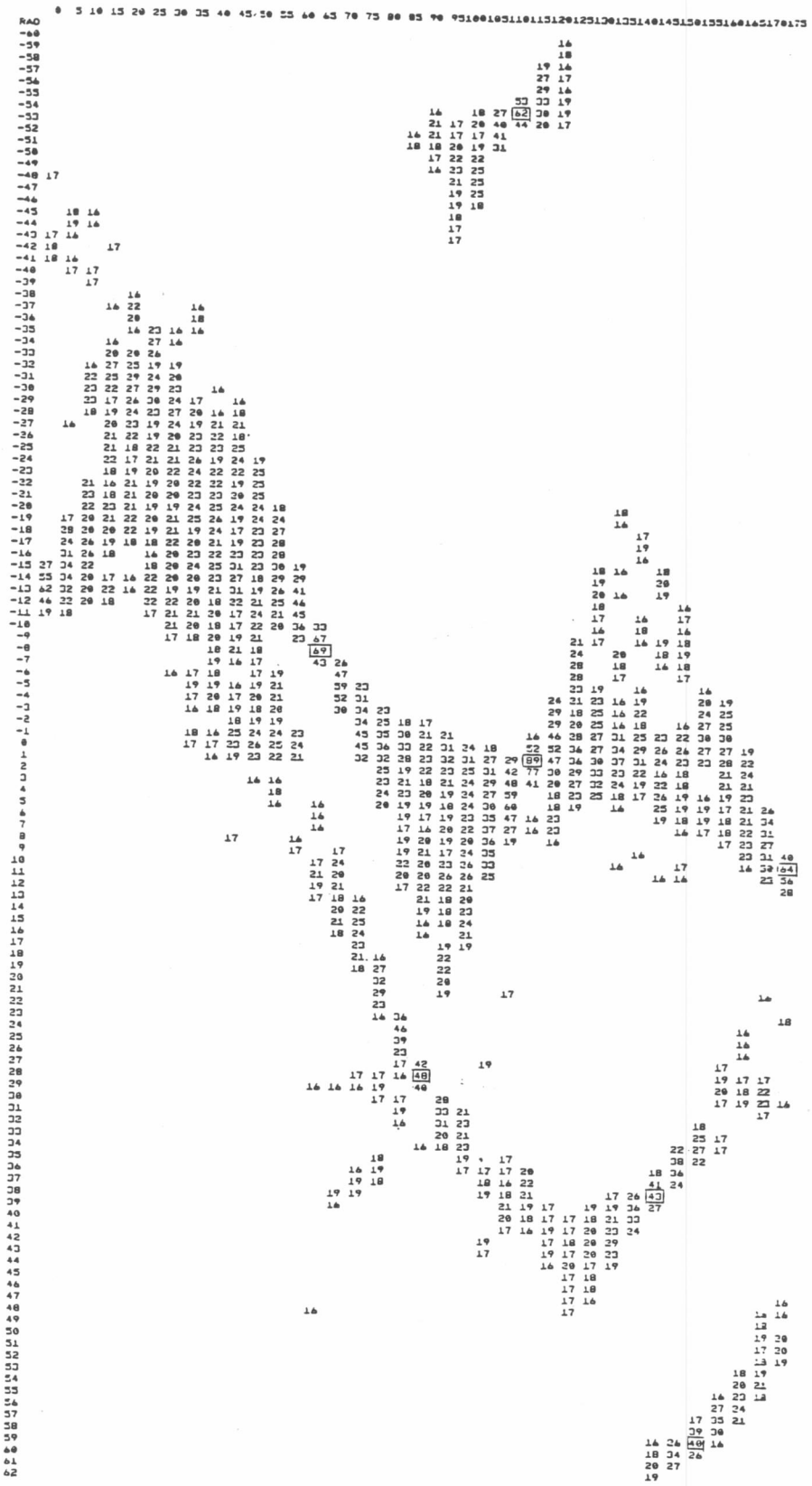


Fig. 9. The cell array for Fig. 8. Squared values are locally maximum and larger than 40.

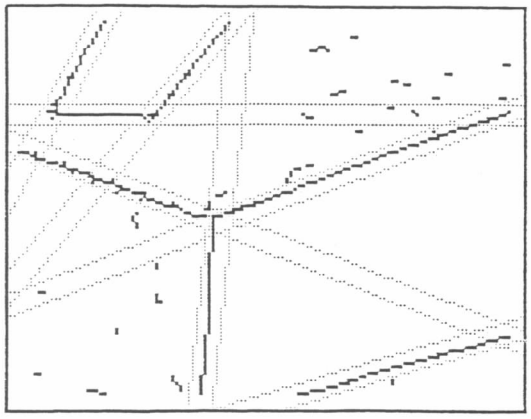


Fig. 10. The edge points of the detected lines in the image plane.

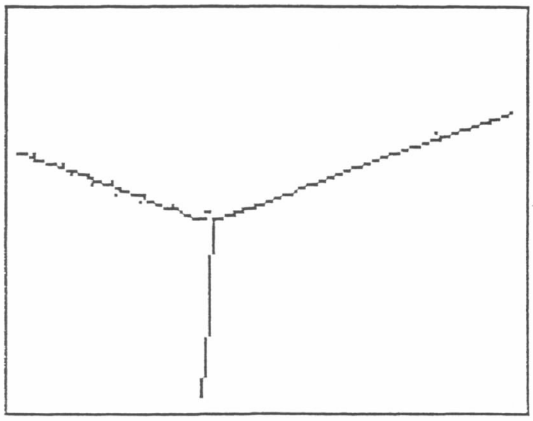


Fig. 11. Identified three corner lines from Fig. 10.

and the square error E^2 reduces to $E^2 = \Sigma(-x_i' \cos\alpha - y_i' \sin\alpha)^2$. E^2 can then be differentiated and the result set zero to get the following solution for α (details omitted):

$$\alpha = \tan^{-1}[(A - \sqrt{A^2 + 4B^2})/(2B)],$$

where

$$A = \Sigma(y')^2 - \Sigma(x')^2,$$

$$B = \Sigma x'y'.$$

Finally, the other parameter ρ is computed from the constraint as $\rho = x_0 \cos\alpha + y_0 \sin\alpha$.

In Fig. 12(a), we show the result of line fitting and refitting applied to Fig. 11. And in Fig. 12(b), we show the result of superimposing Fig. 11 on Fig. 12(a), from which we see that most of the line points found by the corner line identification have been fitted to three corner lines. It is reminded by the way that the line parameters ρ and α computed above are used in computing the camera location parameters as described in Section 2.

4. SYSTEM CALIBRATION

Location precision is the main concern in this study, therefore, much effort has been put into system calibration, including image distortion correction, camera focus length calibration, and image center location.

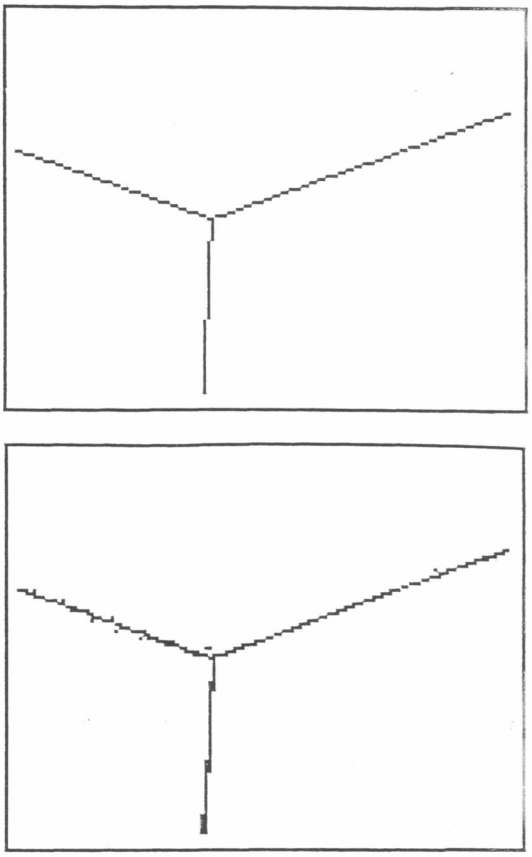


Fig. 12. (a) The result of LSE fitting and refitting, (b) the result of superimposing Fig. 11 on Fig. 12 (a).

4.1. *Image distortion correction*

Imaging devices usually cause geometric distortion in the taken images. Figure 13 shows an image of a grid pattern in which severe geometric distortion is found at the upper edge of the image. Such distortion can be corrected by first finding a linear piecewise transformation which maps every three grid points into their distorted positions in the image, and then applying the transformation as well as bilinear interpolation to correct other images into their nondistorted versions[12]. Figure 14 is the result of distortion correction applied to Fig. 13. For details, see Ref. (11).

4.2. *Calibration of camera focus length*

The calibration of the camera focus length is accomplished by using the principle of similar triangles in the following way. Let the known length of one side of each square in grid pattern and the observed side length of the square in the grid image be w and k, respectively. Also, let the distance from the camera lens center to the grid pattern (pasted on a wall) be d. Then, the camera focus length f, defined as the length from

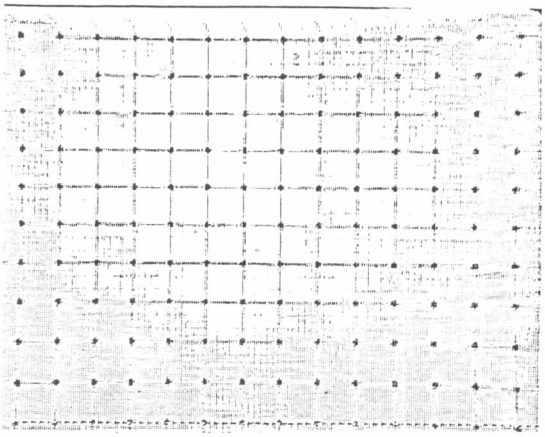


Fig. 13. Grid pattern showing geometric distortion.

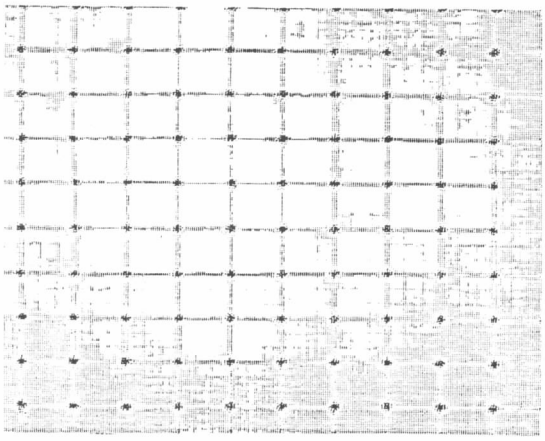


Fig. 14. The result of distortion correction with each grid moderately enlarged.

the lens center to the image plane, is easily derived to be $f = kd/w$. Note that the focus f is used in computing the camera parameters as described in Section 2.

4.3. *Calibration of center of image plane*

The center of the image plane, which is defined as the intersection of the image plane and the camera optical axis, is used as the origin of the image coordinate system. Inaccuracy of its position will affect the coefficients of the equations of the three corner lines, and so the computed camera location. The idea behind the approach we use for the calibration of image plane center is the principle of the vanishing point.[7] As shown in Fig. 15, a black rectangular shape is attached to a wall which is arranged to be parallel to the camera optical axis. The rectangle is half above the height of the camera optical axis. Since the upper and the lower sides of the rectangle are both arranged to be parallel to the optical axis, their extended lines in the image (shown in the upper left corner of Fig. 15) should intersect, according to the vanishing point principle, at the point which is just the center of image plane. Figure 16 (a) shows an image of the rectangular shape, and Fig. 16(b) shows the image processing result of vanishing point finding.

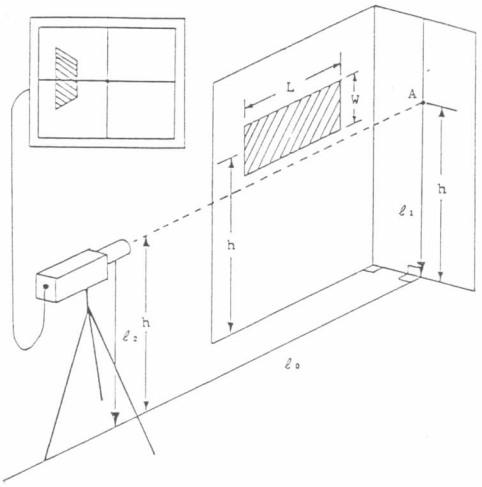
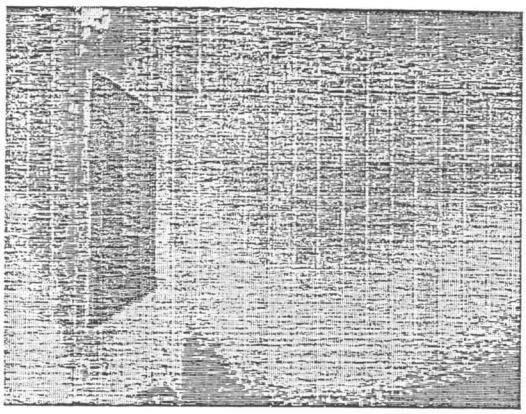


Fig. 15. The arrangement for calibration of the image center point.



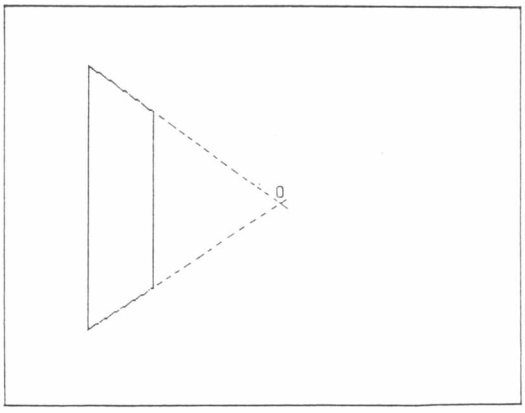
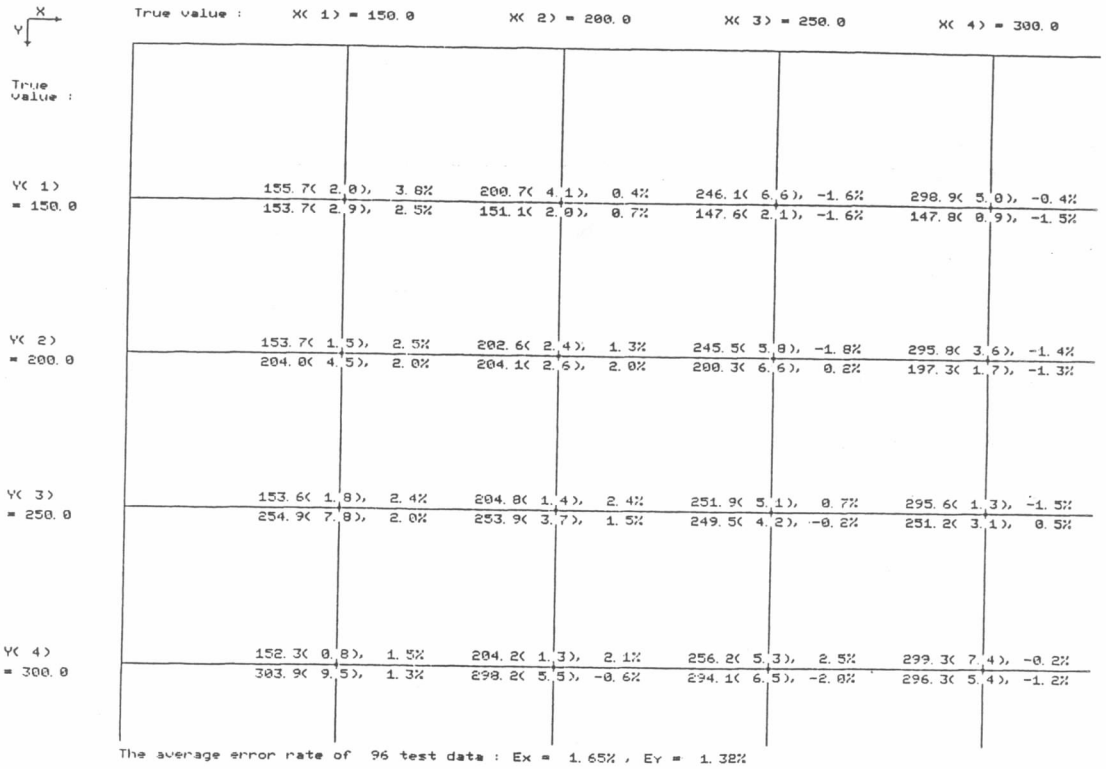


Fig. 16. Calibration of the center of image plane. (a) An image of the rectangle, (b) the image center point found at location O.

5. EXPERIMENTAL RESULTS AND ERROR ANALYSIS

The system we use for experiments is a Matrox imaging system connected to a DEC VAX-11/780 minicomputer host. In a house with its ceiling about two meters high from the camera, the image of a house corner is taken at 16 different locations (i.e. different camera positions and directions), and six times at each location. Each corner image is processed and corresponding camera parameters computed according to the equations derived in Section 2. At each location, the real camera position (X_c, Y_c) is measured manually as the reference. The six computed results of the camera position at each location are averaged, and compared with the real position to obtain the error percentage at each location. The standard derivation of the six computed positions is also computed. All the experimental results are shown in Fig. 17(a); the meanings of the numerical data are explained in Fig. 17(b). From the figure, we see that the average error percentages are all less than 5%. Each location takes time about four seconds on the VAX-11/780 using the FORTRAN language, not including the image transmission between the Matrox and the VAX. A computer simulation has additionally been performed to analyze the magnitude of the relative errors introduced by uncertainties in the equation coefficients of the corner lines which are computed using image and numerical analysis techniques. The analysis[11] results in too many graphic diagrams to be included



(a)

$$\frac{m_x(\sigma_x),\ e_x}{m_y(\sigma_y),\ e_y}$$

m_x (or m_y) : mean of x (or y) distance

σ_x (or σ_y) : standard derivation

e_x (or e_y) : error percent rate

(b)

Fig. 17. The experimental result of robot location. (a) Experimental results of 16 robot positions [data meanings shown in (b)], (b) the meaning of the numerical data in (a).

here. We only mention the main conclusion that when the camera is located closer to the Y–Z (or X–Z) plane, the computed Y_c (or X_c) parameter is more easily affected by the errors in the coefficients α_i of the equations $\rho_i = x \cos \alpha_i + y \sin \alpha_i$ of the corner lines, as is intuitively expected. It is also observed that the error found in the image plane center location does not affect the computed camera position (X_c, Y_c) so much as the error found in the line equation coefficients α_i. This means that more attention should be paid to geometric distortion correction than to image center calibration.

6. CONCLUSIONS AND SUGGESTIONS

A simple and practical approach to in-house robot location is proposed in this paper. The approach utilizes more visible house corners as the standard mark instead of artificially-created land objects. A monocular corner image is adequate for each location. In addition to being able to compute robot or camera positions, the viewing angles can also be obtained. The computation involves a lot of triangular and square functions, but their values can be obtained by table lookup if improvement on computation speed is concerned. Assembly languages can also be used. Thus, reduction of location time to within one second using the DEC VAX-11/780 computer is possible.

On the other hand, the location accuracy is found within 5% on the average. Improvement on this can be directed to using better imaging devices, performing more accurate distortion correction and focus length calibration, adopting subpixel image processing and numerical analysis, etc. Further research may include mounting the camera onto mobile robots or autonomous vehicles for real applications, generalizing the method for other house corner structures, and constructing special-purpose hardware for fast location, etc.

SUMMARY

A new approach to robot location in an in-house 3-D space using house corners as the standard mark is proposed. A monocular image of a house corner is first taken. Image processing and numerical analysis techniques are then applied to find the equations of the three lines going through the corner point. Under the reasonable assumption that the distance from the camera to the ceiling is known in advance, the position of the robot, on which the camera is mounted, is finally uniquely determined according to 3-D imaging geometry. Experimental results with location error less than 5% on the average prove the feasibility of the proposed approach. Error analysis useful for determining location precision is also included.

REFERENCES

1. H. P. Moravec, Visual mapping by a robot rover, *Proc. Int. Joint Conf. on Artificial Intelligence*, Tokyo, Japan, pp. 598–600, August (1979).
2. H. P. Moravec, The Stanford cart and the CMU rover, *Proc. IEEE* **71**, 872–884 (1983).
3. I. Fukui, TV image processing to determine the position of a robot vehicle, *Pattern Recognition* **14**, 101–109 (1981).
4. J. W. Courtney and J. K. Aggarwal, Robot guidance using computer vision, *Proc. IEEE Conf. on Trends and Applications, Automating Intelligent Behavior, Applications and Frontiers*, Gaithersberg, MD, pp. 57–62, May (1983).
5. J. W. Courtney and J. K. Aggarwal, Robot guidance using computer vision, *Pattern Recognition* **17**, 585–592 (1984).
6. M. J. Magee and J. K. Aggarwal, Determining the position of a robot using a single calibration object, *IEEE Conf. on Robotics*, Atlanta, Georgia, pp. 140–149, March (1984).
7. J. D. Foley and A. Van Dam, *Fundamentals of Interactive Computer Graphics*. Addison-Wesley, Reading, MA (1982).
8. I. Sobel, On calibrating computer controlled cameras for perceiving 3-D scenes, *Art. Intell.* **5**, 185–198 (1974).
9. Z. Chen, P. H. Chiu and T. C. Chou, A simplified method for matching sensed planar scenes to reference scenes in a 3D space, *Proc. IEEE Conf. on CVPR*, Washington, D.C., pp. 366–369 (1983).
10. D. H. Ballard and C. M. Brown, *Computer Vision*. Prentice-Hall, Englewood Cliffs, NJ (1982).
11. H. L. Chou and W. H. Tsai, Robot location by house corners, *Technical Report*, Department of Information Science, National Chiao Tung University, R.O.C., in Chinese (1985).
12. A. Rosenfeld and A. C. Kak, *Digital Picture Processing*, 2nd Edn, Vol. 2. Academic Press, New York (1982).
13. K. S. Shanmugan and G. Paul, A fast edge thinning operator, *IEEE Trans. on SMC*. **SMC-12**, 567–569 (1982).
14. R. O. Duda and P. E. Hart, Use of the Hough transformation to detect lines and curves in pictures, *Commun. ACM* **15**, 11–15 (1972).
15. S. A. Dudani and A. L. Luk, Locating straight-line edge segments on outdoor scenes, *Pattern Recognition* **10**, 145–157 (1978).

About the Author—WEN-HSIANG TSAI was born in Tainan, Taiwan, Republic of China, on 10 May 1951. He received the B.S. degree from National Taiwan University, Taipei, in 1973, the M.S. degree from Brown University, Providence, RI, in 1977 and the Ph.D. degree from Purdue University, West Lafayette, IN, in 1979, all in Electrical Engineering.

From 1973 to 1975 he served in the Chinese Navy as an Electronics Officer. From 1977 to 1979 he worked as a research assistant in the Advanced Automation Research Laboratory in the School of Electrical Engineering at Purdue University. From November 1979 to August 1984 he was with the Institute of Computer Engineering at National Chiao Tung University, Hsinchu, Taiwan, Republic of China. He is now a Professor of Computer Engineering, Head of the Department of Information Science and Assistant Director

of Microelectronics and Information Science and Technology Research Center, all at National Chiao Tung University. His current research interests include image processing, pattern recognition, parallel processing, multiprocessor systems and computer vision applications in robotics and automation.

About the Author—HUEI-LIN CHOU was born in Tainan, Taiwan, Republic of China on 25 April 1961. He received the B.S. degree in Electrical Engineering from National Cheng Kung University, Tainan, Taiwan in 1983, and the M.S. degree in Computer Engineering from National Chiao Tung University, Hsinchu, Taiwan in 1985. From 1983 to 1985, he served as a research assistant in the Institute of Computer Engineering at National Chiao Tung University. He is currently serving in the Chinese Army as an officer. His research interests include image analysis, pattern recognition and computer architecture.