AN EFFECTIVE METHOD FOR DETERMINING THE ROBOT POSITION

Din–Chang Tseng , Zen Chen and Jenn–Yee Lin

Institute of Computer Engineering
National Chiao Tung University
Hsin Chu, Taiwan, R.O.C.

ABSTRACT

A procedure for determining the position of a mobile robot in a 3–D space is presented. A single 2–D image of a cubic object is used to derive the position of the robot relative to this cube. The derivation is based on the information of vanishing points of the 2–D perspective projection of the cube. The coordinates of vanishing points are iteratively modified to satisfy the imposed constraints: (1) the edge length of the cube is known, (2) the adjacent edges of the cube are mutually perpendicular, and (3) the projected lines of all parallel edges of the cube converge to a vanishing point. In the experiments, a robot arm is automatically guided using the proposed method to grasp a cube. The error analysis of the experimental results is also reported.

1. INTRODUCTION

For autonomous navigation of a mobile robot in an unconstrained environment, it is important to accurately determine the position of the robot in a 3–D space. If the robot has no vision ability, it only depend on the rotation measure of step motor to determine its position, it will lose its position after many translations and rotations because the step motor and wheels may accumulate errors. Especially, in an uneven ground this method will derive more errors. Lately several vision–based techniques are used for this task such as passive–stereo methods, active–stereo methods, monocular vision methods, etc. The passive–stereo method needs to solve complicated correspondence problem [1]. The active–stereo method needs to analyze several pictures and, therefore, is time–consuming [2–3]. The monocular vision method is simple and rapid, but it does not contain the 3–D raw data for immediate use. In general, navigation problems need complicated 3–D scene analysis, so they preferably need the range images as well as intensity images [4–6]. However, for the indoor navigation problem, the environment is relatively easier to place under control, so we can approach this problem by simplifying the problem and use a single image of a calibration mark (or object) for such a vision guidance.

Fukui [7] first solved the robot vision problem by using a square pattern as a calibration mark with its two diagonals in alignment with the horizontal and the vertical axes of the world coordinate system. The camera lens center and the mark center have to be set at the same height. In addition, the optical axis of the camera must pass through the center of the mark. By the application of image processing technique, the projected lengths of the diagonals are computed. By using the trigonometrical relations, they can determine the 2–D location of the camera with respect to the square mark. Courtney, Magee and Aggarwal [8] extended the Fukui's analysis. They used the same calibration mark as Fukui used, but relaxed the constraint that the camera lens center has to be at the same height as the mark center. Nevertheless, they still must know the height of the camera lens center. On the other hand, Magee and Aggarwal [9] used a calibration sphere with two great orthogonal circles on the surface of the sphere as the two axes of the coordinate system. The optical axis of the camera was aligned to pass through the middle of the sphere. Through the image processing technique, the radius and center of the projected sphere and the centers of the visible portions of great circles painted on the sphere are extracted. They used the plane and spherical trigonometric relations to determine the 3–D position of the camera relative to the sphere.

In this paper we propose a new method to determine the 3–D position of a robot using a single image of a cubic object with more freedom than previous works. This method is based on the information of vanishing points of the 2–D perspective projected cube. In order to correct some inaccurate information, we have defined the accuracy of initial vanishing point. The initial vanishing points are compared according to their accuracies. The coordinates of the most accurate vanishing point is accurately computed by an iterative method and the other vanishing points will be forced to satisfy the imposed constraints: (1) the edge length of the cube is known, (2) the adjacent edges of the cube are mutually perpendicular, and (3) the projected lines of all parallel edges of the cube will converge to a vanishing point. In terms of the coordinates of vertices and vanishing points of the projected cube, we can compute the location and orientation of the camera (or the robot) with respect to the cube.

2. THREE–DIMENSIONAL GEOMETRICAL MODEL AND VANISHING POINT

In this paper we shall assume that the cubes are visible in three faces. We define three visible adjacent edges of the cube as the three principal axes of the world coordinate system, i.e., the x–, y–, z– axes, as shown in Fig.1. Let the x_c–, y_c–, and z_c– axes be the three principal axes of the camera coordinate system (or camera model for short) and the image plane be parallel to the $x_c y_c$–plane at $z_c = q$, and finally the lens center be at the origin. Assume (x_i,y_i,z_i) is a 3–D point and (a_i,b_i) is the corresponding image point. Then we have

$$\begin{cases} a_i = \dfrac{qx_i}{z_i} \\ b_i = \dfrac{qy_i}{z_i} \end{cases}$$

The camera model is given in Fig.2. The transformation between world coordinate system and camera coordinate system can be found in [10].

The determination of vanishing points has been discussed in the literature of scene analysis [11–14]. We know that the perspective projection of any set of parallel lines which are not parallel to the image plane will converge to a vanishing point [10]. The vanishing points of a simple projected cube are shown in Fig.3. Due to quantization error of the image, and photometric and geometric distortion of the camera [15], the projected lines of parallel edges of the cube may not converge to a point, as shown in Fig.4. However we can take the center of the triangular confusion zone as the initial vanishing point, then iteratively correct it, as will be described later. The line for each edge is fitted by a least–square method. For our position determination algorithm we need two types of least square line fitting. The type–1 is the traditional least–square line fitting, and the type–2 is one that forces the line to pass through a fixed point.

3. THREE–DIMENSIONAL POSITION DETERMINATION

A. Deriving 3–D position of the cube from precise 2–D information

(*a*) *Finding directions of edges.* Assume the parametric equation of a line is

$$\begin{cases} x(t) = x_0+at \\ y(t) = y_0+bt \qquad -\infty \le t \le \infty , \\ z(t) = z_0+ct \end{cases}$$

which passes through point (x_0,y_0,z_0) and has direction of the vector $[a,b,c]$. Let $(x(t),y(t),z(t))$ be a point on this line. And let $(u(t),v(t))$ be the projected point of $(x(t),y(t),z(t))$ on image plane. Then we have

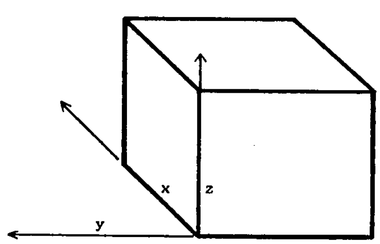

Fig.1. The world coordinate system.

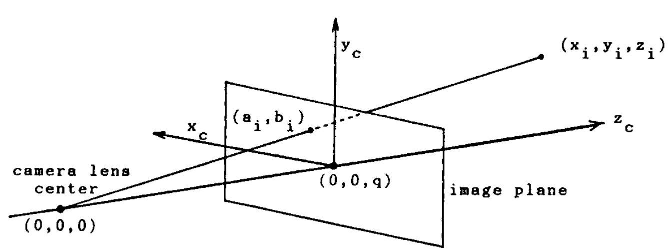

Fig.2. The camera model.

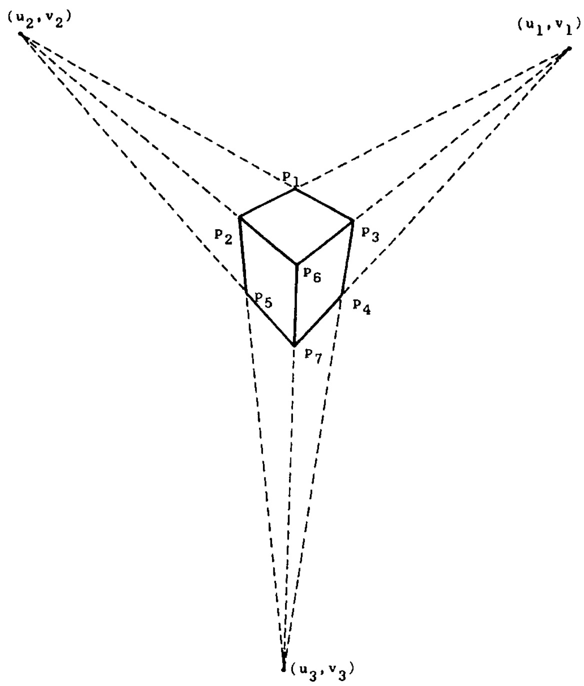

Fig.3. The three-faced cube projection.

$$u(t) = \frac{qx(t)}{z(t)} = \frac{q(x_0+at)}{(z_0+ct)}, \text{ and}$$

$$v(t) = \frac{qy(t)}{z(t)} = \frac{q(y_0+bt)}{(z_0+ct)}.$$

where q is the distance between the camera lens center and the image plane. The vanishing point (u_0,v_0) is obtained as $(u(t),v(t))$ with $t = \infty$. That is

$$u_0 = \lim_{t\to\infty} u(t) = \frac{qa}{c}, \text{ and}$$

$$v_0 = \lim_{t\to\infty} v(t) = \frac{qb}{c}.$$

Thus the direction of this line $[a,b,c]$ is equal to $[u_0,v_0,q]$.

(*b*) *Deriving the planar surface equation.* Assume the directions of two perpendicular edges of a planar surface on the cube are $[u_1,v_1,q]$ and $[u_2,v_2,q]$. Then the normal vector of this planar surface is

$$[u_1,v_1,q]\times[u_2,v_2,q] = (v_1q-v_2q)i+(u_2q-u_1q)j+(u_1v_2-u_2v_1)k.$$

Let $e = v_1q-v_2q$,

$m = u_2q-u_1q$, and

$n = u_1v_2-u_2v_1$.

Then the plane equation becomes

$$ex+my+nz = h, \qquad (1)$$

where h is unknown.

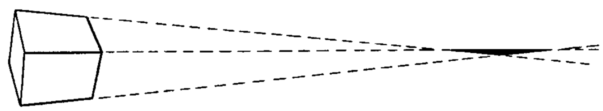

Fig.4. An inaccurate vanishing point. (A triangular confusion zone.)

(*c*) *Determining the 3-D locations of vertices of the cube.* Assume (x_1,y_1,z_1) and (x_2,y_2,z_2) be two adjacent vertices on the planar surface defined by Eq.(1), (a_1,b_1) and (a_2,b_2) be the two corresponding projected vertices on image plane, respectively.

We have

$$(x_i,y_i,z_i) = \left(\frac{a_iz_i}{q},\frac{b_iz_i}{q},z_i\right), \quad i=1,2.$$

Substituting into (1), we obtain

$$e\frac{a_iz_i}{q} + m\frac{b_iz_i}{q} + nz_i = h, \quad i=1,2.$$

It implies that

$$z_i = \frac{h}{\left(\frac{a_ie}{q} + \frac{b_im}{q} + n\right)}, \quad i=1,2.$$

Let $p_i = \frac{a_ie}{q} + \frac{b_im}{q} + n$, $i=1,2$. Then we have

$$(x_i,y_i,z_i) = \left(\frac{a_ih}{qp_i},\frac{b_ih}{qp_i},\frac{h}{p_i}\right), \quad i=1,2.$$

Assume the edge length of the cube is d. Then

$$\sqrt{(x_2-x_1)^2+(y_2-y_1)^2+(z_2-z_1)^2} = d.$$

Thus we have

$$h = \pm d \Big/ \sqrt{\left(\frac{a_2}{qp_2} - \frac{a_1}{qp_1}\right)^2+\left(\frac{b_2}{qp_2} - \frac{b_1}{qp_1}\right)^2+\left(\frac{1}{p_2} - \frac{1}{p_1}\right)^2}.$$

Since z_1 (or z_2) is positive, so the $\pm$ sign of h depends on p_1 (or p_2). That is, h and p_1 (or h and p_2) have the same sign.

B. Improvement of accuracy for noisy image

In order to make sure that the 2-D information is precise, we now present an iterative method to correct coordinates of projected vertices and vanishing points and then compute the accurate 3-D position of the cube.

(*a*) *Use more accurate vanishing point to recover less accurate vanishing point.* There are two principal strategies in this iterative method. The first strategy is using more accurate vanishing point to recover less accurate vanishing points based on the constraints of the cube. Assume the projected cube has three vanishing points. Let (u_1,v_1), (u_2,v_2) and (u_3,v_3) be the three initial vanishing points. Assume coordinates of three vertices of the i–th triangular confusion zone are (u_{i_1},v_{i_1}), (u_{i_2},v_{i_2}) and (u_{i_3},v_{i_3}). Let

$$\begin{cases} \Delta u_i = \max\,(|u_i-u_{i_1}|,|u_i-u_{i_2}|,|u_i-u_{i_3}|) \\ \Delta v_i = \max\,(|v_i-v_{i_1}|,|v_i-v_{i_2}|,|v_i-v_{i_3}|) \end{cases} \quad i=1,2,3. \quad (2)$$

Let $d_i = \Delta u_i^2+\Delta v_i^2$, i=1,2,3. The most accurate vanishing point is defined to be the one with the smallest d_i value.

Assume (u_1,v_1) is the most accurate vanishing point, and then (u_2,v_2) and (u_3,v_3) in sequence. Refer to Fig.3. For a fixed value of (u_1,v_1), (u_2,v_2) and (u_3,v_3) are to be recovered. We know that adjacent edges of the cube are mutually perpendicular. So we have

$$[u_1,v_1,q]\cdot[u_2,v_2,q] = 0.$$

If $\Delta u_2 < \Delta v_2$, then v_2 can be expressed by u_1, v_1, u_2 and q, i.e., $v_2 = -(u_1u_2+q^2)/v_1$, v_2 is a function of u_2. Thus the modification of (u_2,v_2) becomes one–dimensional, that is, the coordinates of (u_2,v_2) are modified along a straight line. Now we iteratively modify u_2, re–compute the coordinates of the projected vertices via the type–2 line fitting formula, compute these coordinates in 3–D formulas, as described by the formulas given in section 3.*A*, until the computed distance between two adjacent vertices of the cube approaches to the known edge length of the cube.

After (u_1,v_1) and (u_2,v_2) are fixed, we can compute (u_3,v_3) based on

$$[u_1,v_1,q]\cdot[u_3,v_3,q] = 0 \text{ and}$$

$$[u_2,v_2,q]\cdot[u_3,v_3,q] = 0\,.$$

(*b*) *Iterative adjustment of the most accurate vanishing point.* The second strategy is to use coarse and fine modification to modify the coordinates of the most accurate vanishing point.

Assume (x_{ij},y_{ij}), $j=1,2,\cdots,7$ are initial projected vertices of the cube on image plane, and (x_{oj},y_{oj}), $j=1,2,\cdots,7$ are the computed projected vertices, as obtained through vanishing points. Let

$$S = \sum_{j=1}^{7} \sqrt{(x_{ij}-x_{oj})^2 + (y_{ij}-y_{oj})^2} \quad (3)$$

be the sum of distance errors between these two sets of projected vertices. S is used to define the stopping criterion of the coarse modification. Each given position of the most accurate vanishing point is associated with such an error. The optimal location of the most accurate vanishing point is defined as the one with the minimal error.

i) Coarse modification. We have found that there may be some local optimal vanishing points in the neighborhood of the initial vanishing point. In order to find a globally optimal solution, we first use a large step size of the iteration (i.e., a coarse modification) around the initial vanishing point to find the best vanishing point, then use a fine modification to approach the final vanishing point. In the coarse modification stage we take a spiral path to approach the best vanishing point. The following diagram represents such a path,

•	17	11	16	•
18	7	3	6	15
12	4	0	2	10
19	8	1	5	14
•	20	9	13	21

where position i denotes the i–th coordinates. The distance between two adjacent coordinates in the spiral pattern is dependent on the size and the shape of the triangular confusion zone. Assume (u_1,v_1) is the most accurate vanishing point. Let Δu_1 and Δv_1 be given in Eq.(2); the step sizes are $\Delta u_1/4$ in the u direction and $\Delta v_1/4$ in the v direction.

ii) Fine modification. The procedure enters the fine modification stage as soon as an error S is less than the stopping criterion of the coarse modification, say, T_2. In our experiment T_2 is set to be 25 pixels. We assume a limit on the number of iterations to avoid infinite computation. In our experiment this number is 360. If all errors on the spiral path are no less than T_2, then take a point with the least error as the initial point of the fine modification. In the fine modification stage, we gradually reduce the step size until it is smaller than 0.1 pixel. At the end, we compute the final error. If the final error is smaller than a value T_4 (in our experiment T_4 is set as 16 pixels), then we accept the result and compute the 3–D locations of all visible vertices of the cube; otherwise, the method fails and a better picture is to be taken for a second analysis.

C. *The algorithm*

Now we use three–faced cube projection as an example to give a brief algorithm to summarize the above process:

ALGORITHM VISION.

Step 1. Use the type–1 line fitting formula to determine the initial projected vertices on image plane.

Step 2. Find initial vanishing points.

Step 3. Determine the most accurate vanishing point. Assume (u_1,v_1) is the most accurate vanishing point, and (u_2,v_2) and (u_3,v_3) follow in order.

Step 4. Use (u_1,v_1) to recover (u_2,v_2), and then use (u_1,v_1) and (u_2,v_2) to compute (u_3,v_3).

Step 5. Use the type–2 least square line fitting formula to obtain the new lines. Find the new projected vertices, and then calculate the sum of distance errors S.

Step 6. (Coarse modification)
If S is greater than the stopping criterion of the coarse modification T_2, then use the coarse modification to modify (u_1,v_1) and go to Step 4. Otherwise, go to Step 7. If the number of coarse modifications is greater than a given number, then select a vanishing point with the smallest error and go to Step 7.

Step 7. (fine modification)
Use the fine modification to modify (u_1,v_1) and execute Step 4 and Step 5. If the step size is greater than a prespecified value, then repeat the fine modification. Otherwise, go to Step 8.

Step 8. If the final error is less than a threshold T_4, then accept the result and compute 3–D locations of all visible vertices. Otherwise, take a new picture to restart.

4. EXPERIMENTAL RESULTS AND DISCUSSIONS

Based on the proposed method, we conduct several experiments. In order to demonstrate the reliability of the experimental results, we control a Mitsubishi RM–101 robot arm to grasp the cube. In the experiments we use a CCD camera with a f = 25 mm lens to take several pictures; the resolution of all images is 256×240 pixels. The cubes have three different edge sizes, 5 cm, 10 cm and 15 cm. The cubes are placed in front of the camera at a distance about from 50 cm to 200 cm. The cubes can be placed at any location with an arbitrary orientation. However, we report only two sets of three–faced cube images with different camera tilt angles. In these experiments three cubes are used and four pictures of each cube are taken. Initially the cube is arbitrarily placed and we take a picture to obtain the first sample, then we rotate the cube by 15^0 around a edge passing through vertices p_6' and p_7' which are the 3–D points corresponding to p_6 and p_7 in Fig.3 and get the second sample. We continuously rotate the cube by 15^0 each time to get the third sample and the fourth sample. For each sample, we compute the distance between the camera lens center and vertex p_7' of the cube and the cube rotation angles. The results are shown in Table 1.

From Table 1, we see that all computed distance errors are less than 1.6%, and all computed rotation angle errors are less than 2 degrees except one case. We also show the iteration numbers for coarse and fine modifications in the table to indicate the speed. There are four iteration numbers of coarse modification that are greater than 200, and one iteration numbers of fine modification that are greater than 200. The main reason for these cases is that the step size of the modification is small, compared to the distance error needed to be corrected.

In addition to the above error analysis we also superimpose the computed projected cubes on the initial projected cubes to inspect the corrections, as shown in Fig.5.

5. CONCLUSIONS

In this paper we have proposed a new method to determine the 3–D position of a mobile robot in a 3–D space. A single view of a cube is used to derive the position of the robot with respect to the cube based on the information of vanishing points of the 2–D perspective projection of this cube. The coordinates of vanishing points can be accurately computed by an iterative modification process. Several experiments have been reported to demonstrate the usefulness of this procedure and the results are satisfactory.

REFERENCES

[1] S. T. Barnard and W. B. Thompson , " Disparity analysis of images ," IEEE Trans. Pattern Anal. Machine Intell, Vol. PAMI–2, No.2, pp.333–340, Jun. 1980.

[2] R. A. Jarvis, "A perspective on range finding techniques for computer vision," IEEE Trans. Pattern Anal. Machine Intell., Vol.PAMI–5, No.2, pp.122–139, Mar. 1983.

[3] J. L. Posdamer and M. D. Altschuler, "Surface measurement by space–encoded projected beam systems," Computer Graphics and Image Processing 18, pp.1–17, 1982.
[4] Y. Yakimovsky and R. Cunningham, "A system for extracting three–dimensional measurements from a stereo pair of TV cameras," Computer Graphics and Image Processing 7, pp.195–210, 1978.
[5] H.P.Moravec, "The stanford cart and CMU rover," Proc. IEEE, Vol.71, No.7, pp.872–884, Jun. 1983.
[6] R. A. Lewis and A. R. Johnston, "A scanning laser rangefinder for a robotic vehicle," in Proc. 5th Int. Joint Conf. Artificial Intelligence, pp.762–768, 1977.
[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] J.W.Courtney, M.J.Magee, and J.K.Aggarwal," Robot guidance using computer vision ," Pattern Recognition, Vol.17, No.6, pp.585–592, 1984.
[9] M. J. Magee and J. K. Aggarwal, "Determining the position of a robot using a single calibration object," in Proc. IEEE Conf. on Robotics, pp.140–149, 1984.
[10] J. D. Foley and A. Van Dam, Fundamentals of Interactive Computer Graphics, Addison–Wesley, Reading, Mass., 1982.
[11] R. O. Duda and P. E. Hart, Pattern Classification and Scene Analysis, Wiley, New York, 1973.
[12] R. M. Haralick, "Using perspective transformations in scene analysis," Computer Graphics and Image Processing 13, pp.191–221, 1980.
[13] H. Nakatani, S. Kimura, O.Saito, and T.Kitahashi, "Extraction of vanishing point and its application to scene analysis based on image sequence ," in Proc. IEEE 5th Int. conf. Pattern Recognition, pp.370–372, 1980.
[14] M.J.Magee and J.K.Aggarwal, "Determining vanishing points from perspective images," Computer vision, Graphics, and Image Processing 26, pp.256–267, 1984.
[15] L. Matthies and S. A. Shafer, "Error modelling in stereo navigation," Carnegie–Mellon Univ, Pittsburgh, PA, Tech. Rep. CMU–CS–86–140, 1986.

Table 1. The estimation results of the three–faced cube projections. (a) when the camera has a tilt angle equal to 15^0. (b) when the camera has a tilt angle equal to 45^0.

(a)

edge length of cube	measured distance	sample labels, computed distances, cube rotations and iteration numbers for coarse and fine modifications			
50	511.14	A11	A12	A13	A14
		510.86*	505.98	507.60	503.02
		0.0**	14.627	29.861	44.039
		12†	1	1	104
		20††	16	19	14
100	934.29	A21	A22	A23	A24
		926.98	919.60	925.28	923.18
		0.0	14.537	30.236	45.028
		253	279	4	22
		21	36	17	14
150	1889.77	A31	A32	A33	A34
		1885.82	1893.27	1889.17	1876.68
		0.0	14.356	31.949	45.799
		9	41	1	1
		15	15	12	9

(b)

edge length of cube	measured distance	sample labels, computed distances, cube rotations and iteration numbers for coarse and fine modifications			
50	528.43	B11	B12	B13	B14
		525.58	533.08	530.36	528.01
		0.0	15.035	29.775	44.972
		38	225	215	2
		13	21	7	18
100	910.48	B21	B22	B23	B24
		910.73	905.31	906.63	924.10
		0.0	14.954	29.046	42.918
		28	18	101	15
		10	16	5	10
150	1973.91	B31	B32	B33	B34
		1978.65	1977.75	1973.27	1955.62
		0.0	13.976	29.362	45.181
		1	1	7	10
		12	226	9	7

*the computed distance,
**the computed rotation,
†the iteration numbers used in the coarse modification, and
††the iteration numbers used in the fine modification.

The units of distances and edge lengths are in mm, and the unit of rotations is in degree.

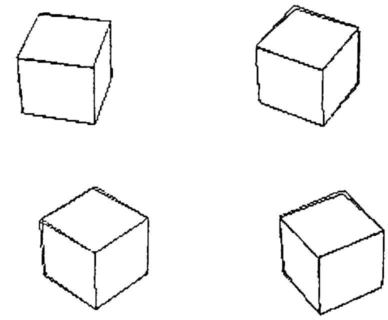

Fig.5. The computed projected cubes are superimposed onto the initial projected cubes to inspect the goodness of results.