Mohamed Bénallal [1,2] and Jean Meunier [2]
[1] École des Mines de Paris 60 Bd Saint Michel F-75272 Paris Cedex 06 France
[2] Université de Montréal D.I.R.O. CP 6128 Centre Ville Montréal (Québec) H3C 3J7
{benallam, meunier}@iro.umontreal.ca

FAST 3D LOCALIZATION OF POLYGONAL OBJECT IN MONOCULAR VISION

Abstract.
We propose an iterative method of 3D localization of 2D or 3D polygonal shapes by monocular vision from a sngle image. The method assumes that the size of the polygonal object is known and that the camera is calibrated. Essentially, the 3D localization is obtained by the resolution of a non-linear system using the parametric equations of the polygonal object (contour) projection in the image plan. Then we show that it is possible with this approach to implement a simple and robust 3D localization process in real time. One interesting application we are currently working on, is the 3D real-time localization of road signs from a mobile vehicle with a monocular vision system.

keywords : 3D localization, Pose problem, CG method for nonlinear system.

1 INTRODUCTION

In this paper, the problem of localization consists in determining the translation and rotation parameters of an object with respect to a coordinate system (e.g. camera) from image(s) knowing the camera intrinsic parameters. In stereovision or with multiple images, this problem is solved by finding landmark correspondences in the two (or more) images knowing the extrinsic parameters of the cameras. In monocular vision with a single image, we must rely on a priori knowledge (to compensate for the lack of information) about the object shape to perform the same task. Both techniques are used in many applications such as car or robot localization [2-7].

The proposed method was realized with the aim of supplying data on distances (and orientations) of polygonal forms (essentially road signs) contained in a road scene (figure 1) in real time. Indeed, this is a question of giving to the driver quantitative distance information about the outside environment for a vehicle moving at a speed of about 50km/hour [8]. These polygonal forms have at least 3 corners (vertices) that can be extracted generally without too much problems using edge detection [3, 10, 11] and simple color segmentation (figure 1).

In this paper we assume that edges and vertices are detected with an appropriate algorithm (Susan edge detector [10] in figure 1).

Figure 1 : (Left) Typical example of a road scene with a polygonal shape (stop sign), with edge detection [10] and color segmentation (right).

In the next section, we focus on the formulation of localization with the polygon parametric equations.

2 CAMERA-POLYGON SYSTEM MODELING

Our goal is to find the distance between a polygon and the camera. In order to do this we choose the coordinate system of the camera where it is easy to establish the relationship between a 3D point with coordinates ($\boldsymbol{x_c, y_c, z_c}$) and its projection in the image plane ($\boldsymbol{u,v}$) with the following equations :

$$u = u_0 + \frac{f}{p_x} \cdot \frac{x_c}{z_c}, \qquad v = v_0 + \frac{f}{p_y} \cdot \frac{y_c}{z_c} \tag{1}$$

where ($\boldsymbol{u_0, v_0}$) are the coordinates of the image center, $\boldsymbol{f}$ the focal length and $\boldsymbol{p_x, p_y}$ the size of the pixel. These parameters are estimated with an appropriate camera calibration technique such as [9]. For a polygon, the 3D points are typically the vertices of its contour.

In figure 2, an example with a polyhedron is shown where one can observe that the line segments ***ab***, ***bc***, ***cd*** and ***da***, correspond respectively to the projections of the line segments ***AB***, ***BC***, ***CD*** and ***DA***. One can easily generalize this to other polygonal shape.

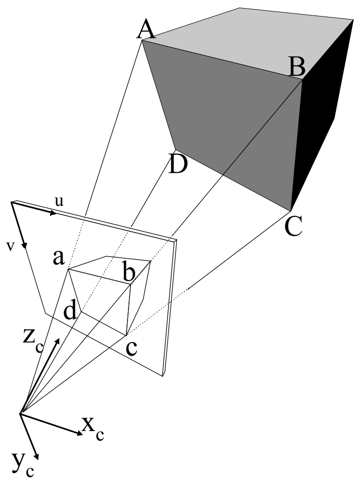

Figure 2 : Projection of a Polygonal shape.

The Euclidean length of each segment that composes a polygonal shape, in the 3D camera coordinate system are obviously the same in any other coordinate systems (invariance). Therefore we will use this knowledge to recover the 3D position of a polygon. For the example of figure 2, these lengths are :

$$AB = \sqrt{(x_A - x_B)^2 + (y_A - y_B)^2 + (z_A - z_B)^2}$$

$$BC = \sqrt{(x_B - x_C)^2 + (y_B - y_C)^2 + (z_B - z_C)^2} \qquad (2)$$

$$CD = \sqrt{(x_C - x_D)^2 + (y_B - y_D)^2 + (z_B - z_D)^2}$$

$$DA = \sqrt{(x_D - x_A)^2 + (y_D - y_A)^2 + (z_D - z_A)^2}$$

Knowing the intrinsic parameters of the camera (calibration) we can calculate the direction cosine for each image points (***a***,***b***,***c***,***d***), projections of (***A***,***B***,***C***,***D***). The line ***OA*** or ***Oa*** is the direction line of the point ***A*** in the camera coordinate system and so on for every points of the polygon. Direction cosines are:

$$\cos\alpha_a = \frac{x_a}{\sqrt{x_a^2 + y_a^2 + z_a^2}}$$

$$\cos\beta_a = \frac{y_a}{\sqrt{x_a^2 + y_a^2 + z_a^2}} \qquad (3)$$

$$\cos\gamma_a = \frac{z_a}{\sqrt{x_a^2 + y_a^2 + z_a^2}}$$

where α, β et γ represents angles between the line ***OA*** with the axes of the camera $\mathbf{x_c}$, $\mathbf{y_c}$ and $\mathbf{z_c}$ (see figure 3).

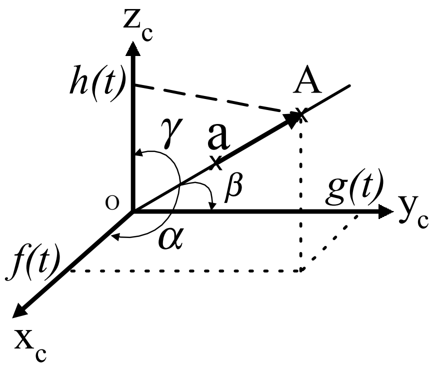

Figure 3 : Direction angles.

A point in a polygon, in the 3D coordinate system, can be represented by its homologue, its direction and ***length*** with the parametric equation :

$$\begin{cases} x_A = x_a - t_a \cos \alpha_a \\ y_A = y_a - t_a \cos \beta_a \\ z_A = z_a - t_a \cos \gamma_a \end{cases} \tag{4}$$

or to simplify the notation (see figure 3):

$$\begin{cases} x_A = f(t_a) \\ y_A = g(t_a) \\ z_A = h(t_a) \end{cases} \tag{5}$$

The Euclidean length of the segment AB of the polygon is:

$$AB = \sqrt{(x_A - x_B)^2 + (y_A - y_B)^2 + (z_A - z_B)^2} \tag{6}$$

This can be rewritten with the parametrical equation as follows:

$$AB = \sqrt{\begin{array}{l} ((x_a - t_a \cdot \cos\alpha_a) - (x_b - t_b \cdot \cos\alpha_b))^2 + \\ ((y_a - t_a \cdot \cos\alpha_a) - (y_b - t_b \cdot \cos\alpha_b))^2 + \\ ((z_a - t_a \cdot \cos\alpha_a) - (z_b - t_b \cdot \cos\alpha_b))^2 \end{array}} \tag{7}$$

By factorization, (t_a and t_b) , we obtain for the segment AB the following equation :

$$\begin{cases} (\cos^2\alpha_a + \cos^2\beta_a + \cos^2\gamma_a) \cdot t_a^2 \\ - 2 \cdot (\cos\alpha_a \cdot \cos\alpha_b + \cos\beta_a \cdot \cos\beta_b + \cos\gamma_a \cdot \cos\gamma_b) \cdot t_a \cdot t_b \\ + (\cos^2\alpha_b + \cos^2\beta_b + \cos^2\gamma_b) \cdot t_b^2 \\ + 2[(x_b - x_a) \cdot \cos\alpha_a + (y_b - y_a) \cdot \cos\beta_a + (z_b - z_a) \cdot \cos\gamma_a] \cdot t_a \\ + 2[(x_a - x_b) \cdot \cos\alpha_b + (y_a - y_b) \cdot \cos\beta_b + (z_a - z_b) \cdot \cos\gamma_b] \cdot t_b \\ - AB^2 + (x_a - x_b)^2 + (y_a - y_b)^2 + (z_a - z_b)^2 = 0 \end{cases} \tag{8}$$

This equation has the form:

$$K_1 \cdot t_a^2 + L_1 \cdot t_a \cdot t_b + M_1 \cdot t_b^2 + N_1 \cdot t_a + P_1 \cdot t_b + Q_1 = 0 \tag{9}$$

with K and M always equal to 1. This is the general quadratic equation of a conic. So by permutation, one can obtain a system of **quadratic** equations (figure 2) :

$$\begin{cases} t_a^2 + L_1 \cdot t_a \cdot t_b + t_b^2 + N_1 \cdot t_a + P_1 \cdot t_b + Q_1 = 0 \\ t_b^2 + L_2 \cdot t_b \cdot t_c + t_c^2 + N_2 \cdot t_b + P_2 \cdot t_c + Q_2 = 0 \\ t_c^2 + L_3 \cdot t_c \cdot t_d + t_d^2 + N_3 \cdot t_c + P_3 \cdot t_d + Q_3 = 0 \\ t_d^2 + L_4 \cdot t_d \cdot t_a + t_a^2 + N_4 \cdot t_d + P_4 \cdot t_a + Q_4 = 0 \end{cases} \tag{10}$$

We can simplify the notation with:

$$\begin{cases} f_1(t_a, t_b, t_c, t_d) = 0 \\ f_2(t_a, t_b, t_c, t_d) = 0 \\ f_3(t_a, t_b, t_c, t_d) = 0 \\ f_4(t_a, t_b, t_c, t_d) = 0 \end{cases} \Rightarrow \quad F(T) = 0 \tag{11}$$

3 CONJUGATE GRADIENT METHOD

In order to get 3D polygon positions in real time, we use the method of the Conjugated Gradient (CG) to solve the quadratic system because it achieves rapid convergence and needs modest storage and computational resources.

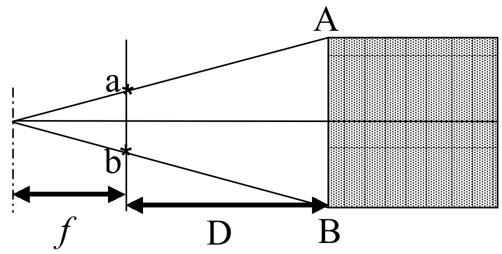

Figure 4 : Similar triangle for estimate the initial vector.

The CG method is probably the most powerful and reliable method in multidimensional. Optimization problem without constraint. The function to be minimized is (equations 10 and 11) :

$$G(T) = F(T)^T \cdot F(T) \tag{12}$$

To initialize the CG algorithm we use the method of similar triangles (see figure 4) and set:

$$t_a^0 = t_b^0 = t_c^0 = t_d^0 = f \cdot \frac{(AB - ab)}{ab} \tag{13}$$

We limited the number of iterations empirically to 40 and we used two stopping criteria:

$$\text{Stop if } \frac{\|T_{t+1} - T_t\|}{\|T_t\|} < \varepsilon \text{ Or stop if } \frac{|F(T_{t+1}) - F(T_t)|}{F(T)} < \varepsilon \tag{14}$$

where ε was set to 10^{-10}.

We choose experimentally Hestenes and Stiefel's version of The CG algorithm [1]. The algorithm has the following form:

with $t_a^0 = t_b^0 = t_c^0 = t_d^0 = f \cdot \frac{(AB - ab)}{ab}$

$T_1 = T_0 - \lambda \cdot \nabla G(T_t)$

$d = -\nabla F(T)$

While $\frac{\|T_{t+1} - T_t\|}{\|T_t\|} > \varepsilon$ or $\frac{|F(T_{t+1}) - F(T_t)|}{F(T)} > \varepsilon$

{

λ that minimize $G(T + \lambda \cdot d)$

$$\beta^{HS} = \frac{\nabla^T G(T_t) \cdot (\nabla G(T_t) - \nabla G(T_{t-1}))}{d_{t-1}^T \cdot (\nabla^T G(T_t) - \nabla G(T_{t-1}))}$$

$d = -\nabla G(T_t) + d \cdot \beta^{HS}$

$T_{t+1} = T_t - \lambda \cdot d$

}

The minimization of λ was carried out with a dichotomy line-search algorithm.

4 EXPERIMENTAL RESULTS

The CCD camera model is TMC-6 from Pulnix (USA) with 752x582 pixels ($\boldsymbol{p_x}$=8.6µm $\boldsymbol{p_y}$=8.3µm) and the lens is a Computar TV lens 1:1.2 12mm (Japan) while the frame grabber (with low-level image processing capabilities) was an Optibase (Israël). The intrinsic parameters of the camera were estimated by a method of camera calibration developed by our group [9] (viewfinder principle). Edge detection was carried out with the Susan edge detector [10] and vertices were selected manually (obviously this step could be well improved but this is not the object of this paper). We tested the algorithm on distances ranging from 2 m to 10 m for a square target of 55 cm/side. Figure 5 shows a comparison of the algorithm results with distances obtained with a laser system that has an accuracy of +/- 30mm. The graph shows the expected linear relationship.

Distance (m)	2 to 4	4 to 6	6 to 8	8 to 10
Error %	0.5%	0.9%	2.3%	4.1%

Table 1 : mean relative errors.

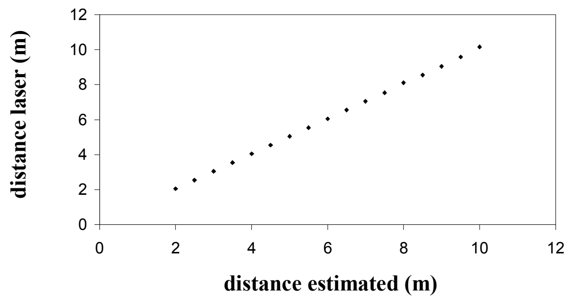

Figure 5 : Laser vs. Algorithm distances

The mean relative error between the distance to the center of the target obtained with the laser and the mean distance of the 4 corners (obtained with our algorithm) is presented in Table 1. One can observe that the error increases with distance mainly due to errors in the computation of the corner positions in the image plane as the projection of the object gets smaller.

The CG algorithm with the Hestenes and Stiefel's coefficient β^{HS} associated to a vector initialization close to the solution allowed us to converge quickly to the solution in real time (t < 20 msec., $\varepsilon = 10^{-10}$ with a Pentium II, 500Mhz, Windows NT SP4).

5 CONCLUSION

We have proposed an iterative method of 3D localization of polygonal shapes by monocular vision from a single image using the parametric equations of the polygonal object. The method simply needs the size of the polygonal object and the intrinsic parameters of the camera. The results are very promising with errors that are essentially due to the edge detection of the polygonal object in the image particularly when the ratio object-distance/object-size increases and the object projection size decreases. Furthermore, one can imagine that the use of the length of diagonals and/or angles between segments will add additional constraints that, although redundant, could improve the results in more difficult situations (noise, large distances…) One interesting application we are currently working on, is the 3D real-time localization of road signs from a mobile vehicle with a monocular vision system

6 REFERENCES

[1] HESTENES M. AND STIEFEL E., "Methods of conjugate gradients for solving linear systems", *J. Res. Nat. Bur. Stand.*, 49 (1952), pp. 409-436

[2] BERTOZZI M., BROGGI A, "GOLD: a parallel Real-Time Stereo Vision System for Generic Obstacle and Lane Detection," *IEEE Transactions on Image Processing* **7**, pp. 62-81, January 1998.

[3] TRUCCO E., VERRI A., *Introductory Techniques for 3D Computer Vision*, Prentice-Hall, 1998

[4] FAUGERAS O. *Three-Dimensional Computer Vision: A Geometric Viewpoint*, The MIT Press, 1993

[5] HARALICK R.M. et al. "Analysis of Solutions of the Three Point Perspective Pose Estimation Problem," *IEEE Int. Conf. On Computer Vision and Pattern Recognition*, pp. 592-598, 1991

[6] BROGGI A. and BERTÈ S., "Vision-Based Road Detection in Automotive Systems: a Real-Time Expectation-Driven Approach," *Journal of Artificial Intelligence Research* **3**, pp. 325-348, December 1995

[7] KOLLER D., MALIK J., LUONG Q.-T., and WEBER J., "An integrated stereo-based approach to automatic vehicle guidance," in *Proceedings of the Fifth ICCV*, pp. 12-20, (Boston), 1995

[8] GRAVILA D.M., PHILOMIN V., "Real time object detection using distance transforms" pp. 274-279 *IEEE International Conference on Intelligent Vehicles*, 1998

[9] BENALLAL M.., MEUNIER J., "A Simple Camera Calibration Method Based on the Principle of the Viewfinder" *Vision Interface 2002*, Calgary, pp. 288-293, 2002

[10] SMITH S.M. and BRADY J.M. SUSAN - a new approach to low level image processing. *Int. Journal of Computer Vision*, 23(1):45--78, May 1997

[11] CANNY J.F. A computational approach to edge detection. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 8(6):679--698, November 1986