Landmark Real-Time Recognition and Positioning for Pedestrian Navigation

Antonio Adán[1], Alberto Martín[1], Enrique Valero[1], and Pilar Merchán[2]

[1] Escuela Superior de Informática. Universidad de Castilla-La Mancha,
13071 Ciudad Real, Spain
{Antonio.Adan,Alberto.Martin,Enrique.Valero}@uclm.es
[2] Escuela de Ingenierías Industriales. Universidad de Extremadura,
06006 Badajoz, Spain
pmerchan@unex.es

Abstract. The aim of this paper is to propose a new monocular-vision strategy for real-time positioning under augmented reality conditions. This is an important aspect to be solved in augmented reality (AR) based navigation in non-controlled environments. In this case, the position and orientation of the moving observer, who usually wears a head mounted display and a camera, must be calculated as accurately as possible in real time. The method is based on analyzing the properties of the projected image of a single pattern consisting of eight small dots which belong to a circle and one dot more at the center of it. Due to the simplicity of the pattern and the low computational cost in the image processing phase, the system is capable of working under on-line requirements. This paper presents a comparison of our strategy with other pose solutions which have been applied in AR or robotic environments.

Keywords: augmented reality, camera pose, landmark, occlusion, real-time.

1 Pose through Perspective Projection Techniques

One of the key points in augmented reality systems for autonomous navigation consists of obtaining an accurate camera pose as quickly as possible. Although there are positioning and tracking systems in controlled environments - for example, technologies based on inertial devices or on networks with a signal receiver/emitter allow 3/6DOF in small environments - for autonomous systems, positioning must be solved with new solutions. Consequently, this issue continues to be an open research field in which innovative solutions are suggested every year.

Depending on each specific application and environment, several factors should be taken into account before choosing the most appropriate technique. The majority of the authors do not make any reference concerning the performance of their method when the landmark is occluded [1], [2], [3], [4], [5], [6], [7], [8]. The authors that take into account occlusion circumstances assume that the landmark is partially occluded but in a non-critical sense. Therefore, those cases correspond to non-severe occlusion. They use natural features [9], [10], [11] or artificial landmarks [12], [13], [14]. Some authors only mention that the system works under occlusion but they do not properly

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 21–28, 2009.

prove that fact ([14]) whereas others ([10], [11]) deal with the problem in depth. In [12], the designed patterns consist of a common vertical barcode and an unique horizontal barcode that distinguish it from any other. In the performed tests, all landmarks are correctly detected in despite of partial occlusions. In [13], several patterns appear in the scene. If one of them is partially occluded, pose estimation can be easily handled by using any non-occluded pattern in the image. In [15], a robust pose method analyzes the geometric distortions of the objects under changes in position. Under real-time requirements, fast image processing may be the key of the general pose approach. There are several authors who provide detailed information concerning this matter [12], [4], [15], [5], [14], [7], [10], [11]. Most of the systems argue to work in real-time conditions: specifically rates are 36-81 fps [12], 30 fps [15], [5], 10 fps [14], 7.3-8.1 fps [7], 15-25 fps [10], [11]. The size of the image, the kind of camera as well as the image processing can have an influence on the final rate of the pose system. For instance, in [4], the system works between 4.2 to 7.2 fps, depending on the kind of camera chosen. In [7], the performance of the system depends on the number and size of potential fiducials in the image.

With regard to the adaptability of the pattern being used in wide distance intervals, most of the referenced methods are designed for use indoors and they seem to work with short and constant ranges.

In any case, very little information is offered in these terms. Exceptions are: [4] where the camera itself establishes a variable range from 0.85 to 1.7 or 3.3 meters and [7] where a tracking procedure works for distances from 50 cm to 5 meters, depending on multi-ring fiducial diameter.

In this paper, the used landmark consists of eight dots corresponding to the vertices of an octahedron and one more double-size dot located at the center of the octahedron. As we will demonstrate in this paper, the system is capable of dealing with severe occlusion of the landmark. Additionally, this landmark allows us to work in a flexible range from 30 centimeters to 7 meters providing similar accuracy than that of most of the referenced approaches. Under realistic conditions, a rate of 30 fps can be achieved. There are no restrictions about the location and orientation of the landmark neither the pose of the camera. Thus the landmark can be set on the floor, ceiling, wall or wherever suitable place.

2 Parameters Definition and Pose Strategy

Before presenting the pose calculation we will outline the framework, the general pose strategy and the parameters to be calculated.

Suppose a human is wearing an AR system composed of a camera integrated into a head-mounted display (see Figure 1) and a laptop in the backpack. The reference systems to be considered are as follows: world reference system (S_w), human reference system (S_h), camera reference system (S_c), image reference system (S_e), computer reference system (S_s) (which is the digital image reference system) and landmark reference system (S_0). Note that the relationship S_w/S_0 is imposed when the landmark is positioned in a specific place and that S_h/S_c is established by the user himself. Moreover, relationship S_c/S_e and S_e/S_s are established by the intrinsic calibration of the camera. As a result, the pose problem is reduced to find the transformation S_0/S_C, which varies as the human moves.

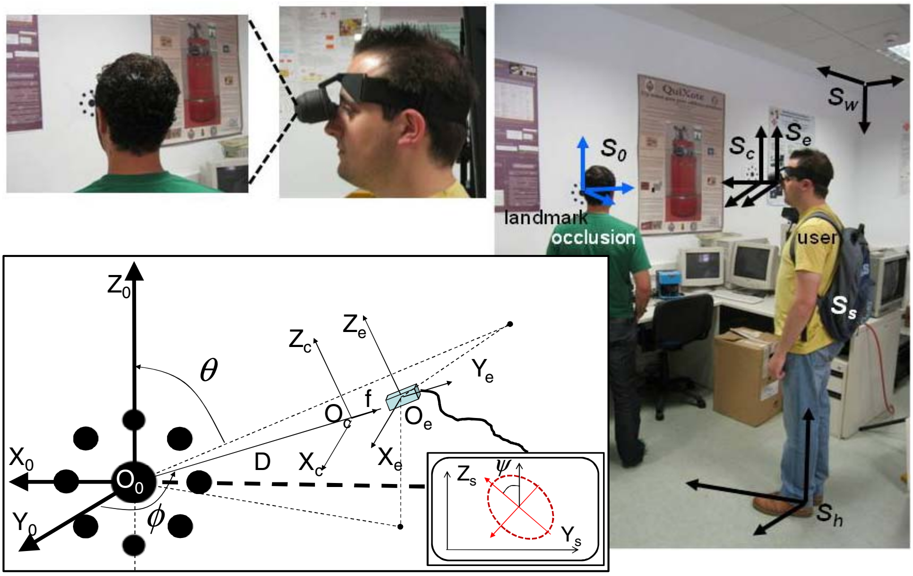

Fig. 1. Right and top) AR based navigation: Reference systems and occlusion of the landmark. Down) Parameters ψ, ϕ, θ and D' in the pattern reference system.

The autonomous procedure presented in this paper is based on the fact that, for any view of the pattern, the outer dots of the pattern belong to an ellipse *E* which changes as the person (camera) moves. Through geometric analysis of *E* and the location of the dots in it, we are able to extract the angular parameters *swing (ψ), tilt (ϕ), pan (θ)* as well as the distance *D'* between the origin of S_0 and the image plane of the camera. From this point forward, we will call the dots (or the center of the dots) P_i , i=1,2....9. (See Figure 1 to consult the pattern reference system and the parameters).

Changes in the position of the user cause changes in ellipse *E*. Thus, variation of *ψ* causes the rotation of the major axis of the ellipse in the image; changes in parameter *ϕ* imply changes in the ellipse eccentricity; when *θ≠0*, dots *{P_1,P_3,P_5,P_7,}* are located outside the axes of the ellipse and, finally, a variation of *D'* makes the length of the major axis of the ellipse change following a quasi-lineal relationship.

Using the camera model presented in Figure 1 down, we establish the transformation between S_0 and S_e reference systems as follows:

$$M = R_{Y_0''}(\psi)\cdot T(D')\cdot R_{Y_0''}(\pi/2)\cdot R_{X_0'}(\phi)\cdot R_{Y_0}(-\theta) \qquad (1)$$

Where the Euler rotations are performed over axes Y_0 , $X_0^{'}$ and $Y_0^{''}$ and *T* corresponds to a translation in axis $Y_0^{''}$.

3 Pose Calculation with Occlusion

As was mentioned in section 1, occlusion circumstances frequently occur in real environments. This pattern has been designed to be used in a wide distance range (from 30 cm to 700 cm away from the user) using the same algorithm. For this reason, it is formed by small circles which in turn belong to an outer circle. Thus, for long distances, the pattern has a set of single marks belonging to a circle whereas for short distances, the pattern is seen as a set of circles. Occlusion is dealt with in this manner.

The pose algorithm with occlusion is established depending on the number of missing dots. We distinguish between several levels of occlusion.

Level 1. When one or two outer dots of the pattern are missing in the image, we categorize it as soft occlusion.

Assume that $(x_{s,i} z_{s,i})$ are the coordinates of the dots viewed in the image and that coefficients C_1, C_2, C_3, C_4, C_5 (taking coefficient $C_6=1$) of the general equation of a conic can be calculated solving the overdetermined system

$$ZH = I \tag{2}$$

where: $H = [\mathrm{C}_1 \quad \mathrm{C}_2 \quad \mathrm{C}_3 \quad \mathrm{C}_4 \quad \mathrm{C}_5]$, $Z = \left[x_{s,i}^2 \quad x_{s,i} z_{s,i} \quad z_{s,i}^2 \quad x_{s,i} \quad z_{s,i}\right]_{i=1,2\ldots6}^T$

Being (x_s, z_s) computer coordinates and I the 1x6 unit matrix. As was previously mentioned, *swing* angle corresponds to the angle (ψ) between the major axis of the ellipse and the vertical reference axis Z_s. *Swing* angle ψ can be easily determined from the estimated parameters , C_1, C_2 and C_3 from equation:

$$\psi = \frac{1}{2} a\,tan \frac{C_2}{C_1 - C_3}, C_1 \neq C_3; \quad \psi = 45^o, C_1 = C_3 \tag{3}$$

The camera-pattern distance can be calculated through the focal and the major axis, (See Figure 2), where a is the major axis of the ellipse the image plane of the camera, f is the focal of the camera and R is the distance from P_9 to whatever external dot. This expression proves that the camera-pattern distance does not depend on the other angular parameters (Figure 2 right).

$$D = \frac{f}{a} R \tag{4}$$

Tilt angle is obtained from the eccentricity of the ellipse following the last equation. Details about obtaining of this parameter can be found in [16].

$$sin\,\phi = e \tag{5}$$

e being the ellipse eccentricity. Therefore, tilt angle is obtained from the eccentricity of the ellipse fitted to the external points of the pattern. Values of ϕ are in the interval [0, 90º]. When $e=0$, $\phi=0$ and the points are fitted to a circle whereas when $e=1$, $\phi=90^o$ and the ellipse is converted into a segment.

Finally, *pan* parameter is obtained through the position of P_1 in the ellipse coordinate system. Let $(x_{e1}, y_{e1}, z_{e1}, 1)_{S_e}$ and $P_1 = (0,0,-R,1)_{S_0}$ be the image coordinates and the pattern coordinates of P_1. Using the transformation M we obtain

$$x_{e1} = -R\cos\theta \tag{6}$$

$$z_{e1} = -R\cos\phi\sin\theta \tag{7}$$

and taking into account that $\dfrac{x_e}{z_e} = \dfrac{x_s}{z_s}$ we finally obtain:

$$\tan\theta = \frac{z_{s1}}{x_{s1}\cos\phi} \tag{8}$$

Equation (3) is maintained. Although dot P_1 was one of the missing dots in the image, equation (8) is maintained for the dual dot P_5 and parameter θ can be obtained. Even if P_1 and P_5 are occluded, we can use both $P_3 = (R,0,0,1)_{S_0}$ and $P_7 = (-R,0,0,1)_{S_0}$ finding $tg\theta = -\cos\phi\dfrac{x_{s1}}{z_{s1}}$.

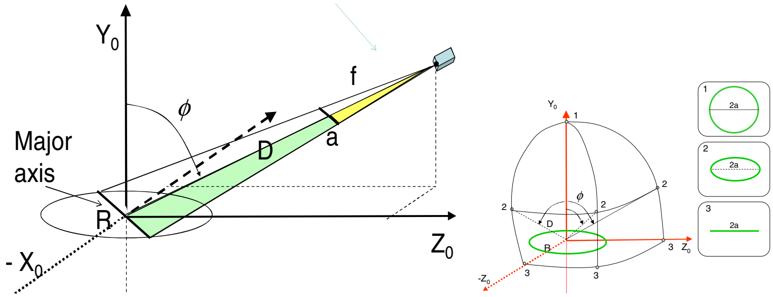

Fig. 2. Analysis of a circle projected in the image and invariance of the major axis length in a sphere around the pattern

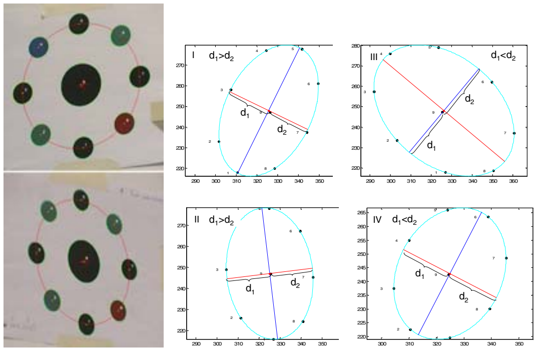

Fig. 3. Left) Images of the landmark from first and second quadrant and ellipses fitted to the dots (in red). Right) Aspect of the fitted ellipse depending on the quadrant in which the camera captures the image.

Table 1. Correction of parameter θ

	d1>d2	*d1<d2*
ψ>0	I	IV
	θ	*θ+π*
ψ<0	II	III
	θ+π	*θ*

When $\psi \neq 0$, non-rotated coordinates x'_s and z'_s must be substituted in the last equation. Equation (8) yields indeterminate values of θ. This problem can be solved by knowing the quadrant (in the system S_0), where the camera is placed. This quadrant is established through the sign of ψ and the position of the point P_9 in the minor axis. It can be proved that, due to projective reasons, P_9 is displaced with respect to the theoretical ellipse center. As a consequence of this, distances d_1 and d_2 from P_9 to the ellipse, in the minor axis direction, are different. Therefore, we can infer whether the pattern is viewed from the left (case $d_1 > d_2$) or from the right (case $d_1 < d_2$). Figure 3 illustrates the displacement of P_9. Table 1 shows the quadrant as well as the applied correction of parameter θ.

Level 2. This happens when more than two outer dots are occluded but the internal dot P_9 is in the image. In this case, equation (2) does not converge and a new strategy must be implemented. This frequently occurs for short user-pattern distances where the view angle of the camera is reduced and a small head movements made by the user can generate loss of the dots in the image. Since any visible circular dot can be viewed as an ellipse, we adapt the pose strategy presented in level 1 for dot P_9.

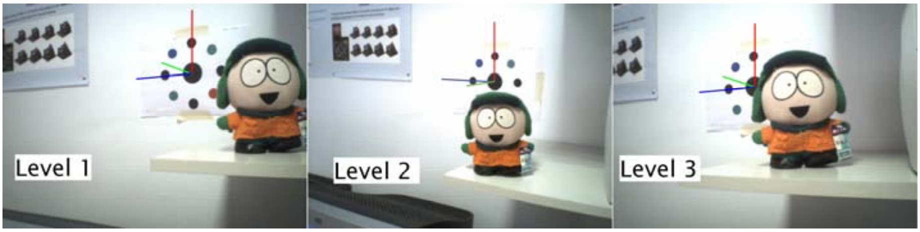

Fig. 4. Examples of several occlusion levels. As we can see, the landmark reference system retroprojected on the image according to the calculated camera positioning.

Level 3. More than two outer dots are occluded and the internal dot P_9 is also missing. This is the highest occlusion level and occurs when the user is near the pattern. After having identified several dots in the image and calculated the pose parameters for each one, we take a weighted mean as the best pose approximation where the weight depends on how close the dot is to the center of the image. Note that in this case, an error is introduced because the pose is calculated in a coordinate system which is translated with respect to S_0. Figure 4 shows examples of different occlusion levels.

4 Experimental Results

In order to prove the applicability of our method under real conditions, we have tested the pose algorithm imposing soft and severe occlusion. The approach was implemented in an autonomous augmented reality system which consists of a Trivisio AR-vision-3D HMD binocular head-mounted display with a color camera and a Quantum3D Thermite portable computer.

A user wearing the portable AR system searches for the pattern when he wants to know its current position in the world coordinate system. Then, the user can see, through one of the two displays of the HMD, the image of the camera and its current position in real-time. The other display is used for superimposing virtual information on the real scene.

The followings phases are repetitively executed on board: I) find/track the pattern in the image, II) segmentation of dots, III) calculate pose parameters. Depending on the occlusion level, we have parameters ψ, ϕ, θ, D' (levels 1 and 2) or parameters for each outer dot ψ_i, ϕ_i, θ_i, D_i' (level 3), IV) obtain a unique pose solution.

The system works with 640x480 images and spends 45 ms to take one frame, process the image and calculate the pose. Thus, the performance average rate is 23 frames/second. Two different environments - indoors and outdoors - have been tested while imposing occasional occlusions. The pattern was occluded by obstacles or people walking in front of the user's viewpoint. Several occlusion circumstances also occurred due to the proximity of the pattern or the user's rapid head movements.

Some information about the performance of the method is included below. Tables 2 and 3 summarizes statistical results of the errors obtained for each estimated user's coordinate and user-pattern distance for both no occlusion and occlusion cases. Absolute and relative errors are presented in two sub-tables. For each case, average, standard deviation, greatest and smallest errors are presented as well. Promising results has been obtained in both cases. Note that position error average was below 5cm for non-occluded case and below 7cm in case of partial occlusions. These results are acceptable enough in the framework we are carrying out in where the user observes the overlaps virtual models from distances always highest than one meter.

We have also designed similar patterns with higher dimension and with a color code which can easily identify each pattern in an extensive environment but this experimental report concerns non-colored patterns and performance under occlusion.

Table 2. Experimental results without occlusion

Abs. Errors (cm)	e(X)	e(Y)	e(Z)	e(D')
Average	2,4	2,33	4,76	2,60
Std.Dev	3,1	1,71	4,51	1,96
Greatest	10,1	6,5	13,6	5,6
Smallest	0,2	0,3	0,1	0,3
R. Errors (%)	e(X)	e(Y)	e(Z)	e(D')
Average	0,60	0,64	6,53	1,52
Std.Dev	0,51	0,45	6,49	1,56
Greatest	1,76	1,42	20,80	5,69
Smallest	0,03	0,06	0,17	0,11

Table 3. Experimental results with occlusion

Abs. Errors (cm)	e(X)	e(Y)	e(Z)	e(D')
Average	6,3	5,9	7,0	5,7
Std.Dev	4,0	3,0	3,7	2,1
Greatest	12,2	10,2	14,1	7,0
Smallest	1,1	0,4	2,0	0,91
R. Errors (%)	e(X)	e(Y)	e(Z)	e(D')
Average	1,31	1.67	4,43	1,98
Std.Dev	1,23	1,66	3,14	1,03
Greatest	3,05	5,02	15,23	3,77
Smallest	0,92	1,02	0,34	0,76

5 Conclusions

The method presented in this paper solves the location problem using a single camera on board an AR reality system. Until now, the majority of based-on-vision pose solutions concern mobile robots applications where the camera is onboard the robot having slow and controlled movements. For augmented reality applications, like ours, the camera is carried over a human head which involves quick and unexpected camera movements. In this sense, we propose a pose method in an unusual environment.

The pose is calculated after analyzing the projected image of an artificial landmark consisting of nine dots. Due to the simplicity of the pattern and the low computational cost in the image processing phase, the system is capable of working under on-line requirements in AR-based navigation applications. Furthermore, opposite to most of the landmark-based positioning solutions, the system works in severe occlusion circumstances and for a wide distance range which makes it more robust than other solutions.

Our approach is being used for AR systems in autonomous navigation for humans yielding excellent results. Experimentation, advantages and restriction of this technique have been illustrated in the paper.

References

1. Cobzas, D., Jagersand, M., Sturm, P.: 3D SSD tracking with estimated 3D planes. Journal of Image and Vision Computing 27, 69–79 (2009)
2. Duan, F., Wu, F., Hu, Z.: Pose determination and plane measurement using a trapezium. Pattern Recognition Letters 29(3), 223–231 (2008)
3. Feng, W., Liu, Y., Cao, Z.: Omnidirectional Vision Tracking and Positioning for Vehicles. In: ICNC 2008. Fourth International Conference on Natural Computation, vol. 6, pp. 183–187 (2008)
4. Fiala, M.: Linear Markers for Robots Navigation with Panoramic Vision. In: First Canadian Conf. Computer and Robot Vision, 2004. Proceedings, pp. 145–154 (2004)
5. Jang, G., et al.: Metric Localization Using a Single Artificial Landmark for Indoor Mobile Robots. In: International Conference on Intelligent Robots and Systems, 2005 (IROS 2005), pp. 2857–2862 (2005)
6. Josephson, K., et al.: Image-Based Localization Using Hybrid Feature Correspondences. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–8 (2007)
7. Neumann, U., et al.: Augmented Reality Tracking in Natural Environments. In: International Symposium on Mixed Reality, 1999. ISMR 1999, pp. 101–130 (1999)
8. Xu, K., Chia, K.W., Cheok, A.D.: Real-time camera tracking for marker-less and unprepared augmented reality environments. Image and Vision Computing 26(5), 673–689 (2008)
9. Se, S., Lowe, D., Little, J.: Mobile Robot Localization and Mapping with Uncertainly using Scale-Invariant Visual Landmarks. The International Journal of Robotics Research 21(8), 735–757 (2002)
10. Vachetti, L., Lepetit, V., Fua, P.: Combining Edge and Texture Information for Real-Time Accurate 3D Camera Tracking. In: Third IEEE and ACM International Symposium on Mixed and Augmented Reality, 2004. ISMAR 2004, pp. 48–56 (2004)
11. Vachetti, L., Lepetit, V., Fua, P.: Stable Real-Time 3D Tracking using Online and Offline Information. IEEE Transactions on PAMI 26(10), 1385–1391 (2004)
12. Briggs, A.J., et al.: Mobile Robot Navigation Using Self-Similar Landmarks. In: IEEE International Conference on Robotics and Automation, 2000. Proceedings. ICRA 2000, vol. 2, pp. 1428–1434 (2000)
13. Kato, H., et al.: Virtual Object Manipulation on a Table-Top AR Environment. In: Proceedings of IEEE and ACM International Symposium on Augmented Reality, 2000 (ISAR 2000), pp. 111–119 (2000)
14. Koller, D., et al.: Real-time Vision-Based Camera Tracking for Augmented Reality Applications. In: ACM Symp. on Virtual Reality Software and Technology, pp. 87–94 (1997)
15. Hager, G.D., Belhumeur, P.N.: Efficient Region Tracking with Parametric Models of Geometry and Illumination. IEEE PAMI 20(10), 1025–1039 (1998)
16. Adan, A., Martín, A., Chacón, R., Dominguez, V.: Monocular Model-Based 3D Location for Autonomous Robots. In: Gelbukh, A., Morales, E.F. (eds.) MICAI 2008. LNCS (LNAI), vol. 5317, pp. 594–604. Springer, Heidelberg (2008)