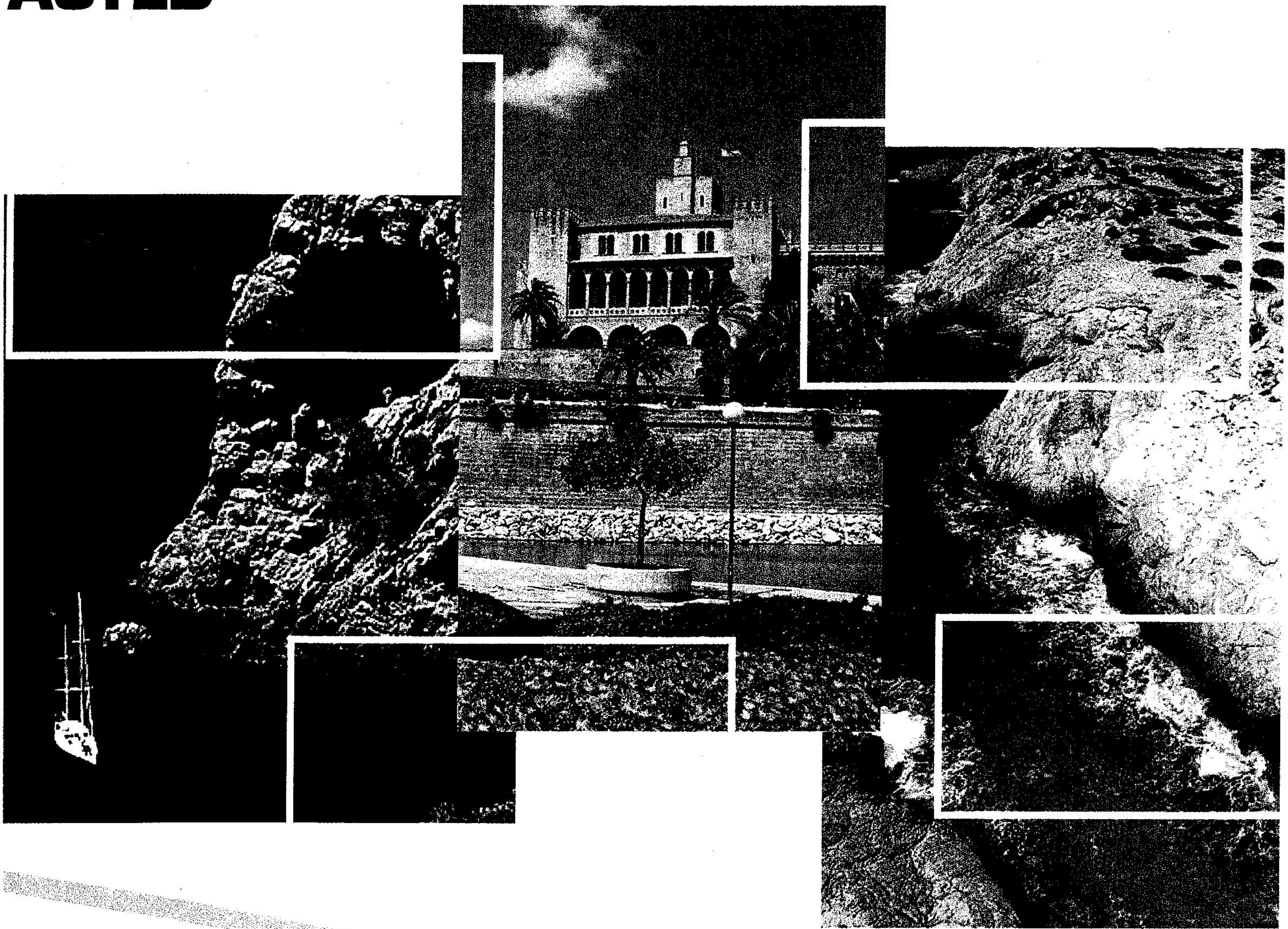
Oscar

ASTED

A Publication of the International Association of Science and Technology for Development



oceedings of the Sixth IASTED International Conference on

VISUALIZATION, IMAGING, AND IMAGE PROCESSING

ɟitor: J.J. Villanueva

August 28 – 30, 2006
Palma de Mallorca, Spain

3BN: 0-88986-598-1
SN: 1482-7921

ACTA Press Anaheim | Calgary | Zurich

Proceedings of the Sixth IASTED International Conference on **Visualization, Imaging, and Image Processing**, held August 28 –30, 2006, Palma de Mallorca, Spain.

SPONSORS

The International Association of Science and Technology for Development (IASTED)

- Technical Committee on Image Processing
- Technical Committee on Computers
- Technical Committee on Visualization

World Modelling and Simulation Forum (WMSF)

EDITOR

J.J. Villanueva – Computer Vision Centre and Autonomous University of Barcelona, Spain

PROGRAM CO-CHAIRS

A. Ebert – University of Kaiserslautern, Germany
A. Sappa – Computer Vision Centre, Spain

KEYNOTE SPEAKER

N. Petkov – University of Groningen, The Netherlands

INTERNATIONAL PROGRAM COMMITTEE

ADDITIONAL REVIEWERS

For each IASTED conference, the following review process is used to ensure the highest level of academic content. Each full manuscript submission is peer reviewed by a minimum of two separate reviewers on the International Program Committee/Additional Reviewers list. The review results are then compiled. If there are conflicting reviews, the paper is sent to a third reviewer.

ACTA Press
P.O. Box 5124
Anaheim, CA 92814–5124
USA

ACTA Press
B6, Suite #101, 2509 Dieppe Ave SW
Calgary, Alberta T3E 7J9
Canada

ACTA Press
P.O. Box 354
CH–8053 Zurich
Switzerland

Publication Code: 541

Proceedings of the Sixth IASTED International Conference
Visualization, Imaging, and Image Processing
August 28–30, 2006, Palma de Mallorca, Spain
ISBN Hardcopy: 0-88986-598-1 / CD: 0-88986-600-7

BEHAVIOUR-BASED MULTI-ROBOT FORMATIONS USING COMPUTER VISION

L. Payá, A. Gil, O. Reinoso, M. Ballesta, R. Ñeco
Departamento de Ingeniería de Sistemas Industriales
Miguel Hernández University. Avda, de la Universidad s/n.
Ed. Torreblanca. 03202, Elche (Alicante), Spain
E-mail: {lpaya, arturo.gil, o.reinoso, ramon.neco}@umh.es

ABSTRACT

Multi-robot systems is a very attractive field of research in robotics, due to the fact that a team of robots is able to accomplish certain tasks that a single robot cannot carry out on his own. This paper presents a behaviour-based method for multi-robot formations using computer vision. A team of robots must move around the environment, keeping a relative position between them and avoiding the collision with possible obstacles. Computer vision has been used for the localization of the robots using the images captured by the camera they carry on their top. Different methods have been tested to obtain the leader robot pose, developing finally a very stable method to calculate distance and orientation. The final control has been built as a composition of several basic behaviours that calculate an action model using the information of the environment. These actions are composed using fuzzy logic to obtain a high-level behaviour. The algorithms developed have been tested over a team of *WiFiBot* mobile robots, with good results.

KEY WORDS

Tracking, multi-robot systems, formations, behaviours, control schema and fuzzy logic

1. Introduction

In the last years, the researching around multi-robot systems has suffered a great develop due to several reasons. A multi-robot system is able to accomplish certain tasks that a single robot cannot carry out. Several revisions around this discipline have been done, collecting different applications of these systems [1] [2] [3].

One of the problems that rise in the field of collaborative robotics is the maintenance of formations, what implies a team of robots to move around keeping a relative position between them, and avoiding the collision with possible obstacles at the same time. Desai *et al.* [4] expose how to carry out this task applying the control theory laws, for example, with the aim of following one or more leaders using a feedback, keeping a distance and orientation previously known and avoiding obstacles. The problem of these controls is the complexity they suppose for structures with a big number of robots. Spletzer *et al.* [5] use this method to pull objects that are contained in the structure. A robust choice to coordinate the movement of the formation is the virtual structures method [6] which deals with the multi-robot system as if it was a rigid solid. It presents a disadvantage in obstacle avoidance, which would be included in a trajectory generator, in a level more deliberative than reactive. Latombe [7] and Arkin and Balch [8] use potential fields to calculate the control actions. There is a potential field on all the points of the space, that will depend of the environment, with the contribution of different attraction fields (destination) and repulsion ones (obstacles). The robot will move looking for minimum potential points. The main drawback is the presence of local minimums in the potential field, which can cause the robot stops. To try to overcome this drawback, a random field could be added. Barfoot and Clark [9] created a system of autonomous trajectories that generates an independent trajectory for each robot knowing the trajectory the formation must go across. They employed a global reference system to estimate the position of each robot and a planner. At last, inspired in biology, in the behaviour-based control proposed in [10], the combination of several basic behaviours constitutes the global behaviour. To carry out the control of formations, a reference must be taken in the formation. Each robot must have a relative position respect this reference, and must include a behaviour that brings it to that position. The major advantage of this method is its simplicity, due to the fact that it splits the problem in different simple behaviours, and the method can be improved easily, just adding new behaviours.

This paper presents a behaviour-based method for multi-robot formations. The objective is that the robots follow a leader, keeping a pre-established distance and orientation. The position of the leader is tracked using computer-vision methods. The remainder of the paper is organized as follows. In section 2, the location system based in computer vision is described. Section 3 shows the platform used to develop the control and the implemented behaviours. In section 4, the experimental results are presented. Finally, conclusions of this work are exposed.

2. Location system

The goal of this system is to obtain the position and orientation of the leader robot $[x_l, y_l, \theta_l]$, using the coordinates system shown of fig. 1 through computer vision techniques. The leader will be identified using a mark on its back.

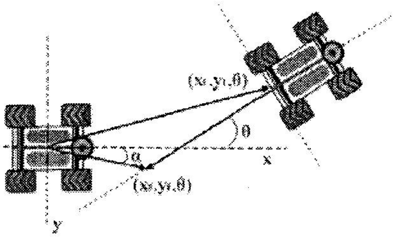


Fig. 1. Coordinates system for the localization of the leader robot.

The follower must previously know the shape and size of the graphics on the label to be able to localize the leader. The mark must contain a minimum of four significant points to avoid ambiguities in the reconstruction of the pose. This way, we design a label with four black points with 2.5 cm of diameter whose centers form a 4 cm side quadrate over white background.

Three methods have been tested to calculate the position and orientation of the leader. The first one is based on the Thales Theorem. Fig. 2 shows how the system acquires the image. Supposing the label is centered on the image plane, the position of the center of the label and its orientation can be estimated studying the proportions between the triangles shown. However, this method presents several disadvantages. The new measure is based on the previous value, so the errors are propagated and not corrected. Also, this method produces a very unstable estimation and presents serious difficulties to estimate the sign of the orientation. More accurate results can be obtained using the principles of projective geometry [11], but with this method, the position depends on the orientation previously calculated so, errors are also propagated. It just works correctly when the calibration is very consistent and the object is situated close to the camera.

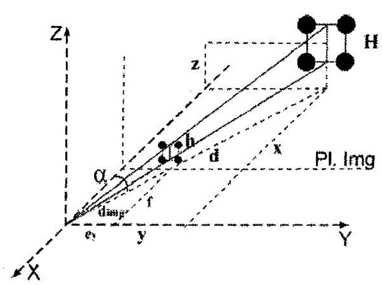


Fig. 2. Formation of the image on the image plane.

Taking into account these facts, a new method has been introduced, combining certain features of the two methods commented. It will let us obtain independently the distance and orientation to the leader in a very stable way. Fig. 2 shows how the image of the four points is formed when they are in an arbitrary situation. The distance to the object can be calculated from the proportions of the triangles on the figure.

$$d = d_{img} \cdot \frac{H}{h} \tag{1}$$

where d is the distance from the camera to the object measured on the xy plane, d_{img} is the distance from the optical centre to the projection of the centre of the object in the image on the xy plane, H is the real height of the object an h its height on the image plane. To transform the distances to pixels the scale factor K_x must be used:

$$d = d_{img} \cdot K_x \cdot \frac{H}{h_{px}} \tag{2}$$

being h_{px} the height of the object on the image plane in pixels. d_{img} will be equal to the focal distance only when the object is perfectly aligned in the centre of the image. In a general situation, as the one shown on fig. 2, for an arbitrary position of the object and assuming that it has just suffered a displacement and no rotation, or that it has only been rotated on the z-axis, d_{img} can be calculated as follows:

$$d_{img} = \sqrt{f^2 + e_y^2} = \frac{1}{K_x} \cdot \sqrt{(f \cdot K_x)^2 + e_{y\,px}^2} \tag{3}$$

where e_y is the distance from the centre of the image to the projection of the object over the y-axis, and $e_{y\,px}$ is the same distance in pixels.

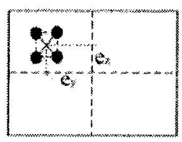


Fig. 3. Coordinates of the center of the object in the image plane.

The angle α, between the x-axis and the line that joins the optical centre with the object on the xy-plane, allows calculating all the desired coordinates:

$$\begin{aligned} \alpha &= \arctan\left(\frac{e_{y\,px}}{f \cdot K_x}\right) \\ X_L &= d \cdot \sin\alpha \\ Y_L &= d \cdot \cos\alpha \\ Z_L &= d \cdot \frac{e_x}{d_{img}} \end{aligned} \tag{4}$$

being e_x the horizontal distance in the image from the centre of the object to the centre of the image. This way, the magnitudes h, e_x and e_y must be measured over the image. To do this, the collineation matrix, obtained from several points in the image, will be used [11]. Previously, a change in the coordinates system will be made so that the homogeneous coordinates on the object plane of the central point are $[0,0,1]^T$. Then, its coordinates in the image plane will be:

$$P_C = C \cdot \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T \tag{5}$$

where C is the collineation matrix. The height can be obtained following the same procedure taking the points $[0, H/2, 1]^T$ and $[0, -H/2, 1]^T$. With this procedure, the position (X_L, Y_L, Z_L) of the central point of the object will be already known. The next step consists on identifying its orientation. To do this, two opposite points P_L and P_R will be used, as shown on fig. 4.

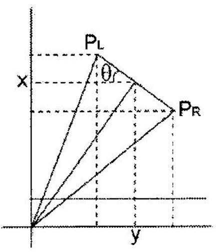


Fig. 4. Points used to calculate the orientation.

The coordinates of these points are:

$$\begin{aligned} P_R &= \left(X_L - \frac{A}{2}\sin\theta_L \quad Y_L + \frac{A}{2}\cos\theta_L \quad Z_L \right) \\ P_L &= \left(X_L + \frac{A}{2}\sin\theta_L \quad Y_L - \frac{A}{2}\cos\theta_L \quad Z_L \right) \end{aligned} \tag{6}$$

Using the Thales theorem on Fig. 2:

$$\begin{aligned} \frac{Y_L + \frac{A}{2}\cos\theta_L}{X_L - \frac{A}{2}\sin\theta_L} &= \frac{e_{YR}}{f \cdot K_x} \\ \frac{Y_L - \frac{A}{2}\cos\theta_L}{X_L + \frac{A}{2}\sin\theta_L} &= \frac{e_{YL}}{f \cdot K_x} \end{aligned} \tag{7}$$

where e_{YR} and e_{YL} are the coordinates in pixels of the projection of the points P_R and P_L onto the image, measured along the x-axis of the image reference system, that is the Y axis of the robot reference system. So, it can be deducted:

$$\sin\theta_L = \frac{2fK_xY_L - X_L\left(e_{YR} + e_{YL}\right)}{\frac{A}{2}\left(e_{YL} - e_{YR}\right)} \tag{8}$$

To obtain e_{YR} and e_{YL} the collineation matrix must be employed again, applying it to the points $[-A/2, 0, 1]^T$ and $[A/2, 0, 1]^T$. This way, the desired position and orientation of the leader are calculated independently in a very robust way. After several experiments, the position measurement presents a reasonable relative error despite the camera is far from the robot. The measure of the orientation is also more robust comparing to the other methods. Fig. 5 shows these results. The Thales method does not offer consistent results when orientation is near to zero and for long distances. The method used (method 3) is the most robust in all cases.

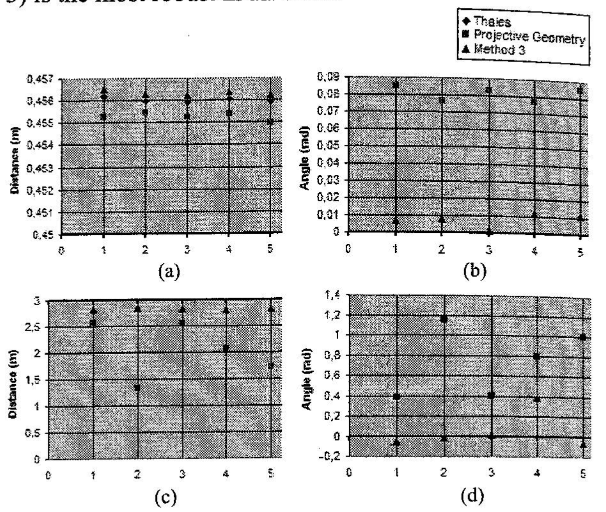


Fig. 5. Distance and angle measured using the three methods for a distance of 45 cm and orientation 0 rad (a) and (b), and for 3 m distance, 0 rad orientation (c) and (d).

3. Implementation on the WiFiBot

3.1 Description of the system

The *WiFiBot* is a low-cost robot characterized by its flexibility. There are two models: *SC* and *4C*. Both of them include an Ethernet camera, two infrared sensors situated in the front part with a reach of 1.2 meters and four optical encoders with 300 sectors to measure the speed of each wheel independently. A PC-104+ board, with *Linux Debian* operative system has been added in both models. Fig. 6 shows both *WiFiBot* models.

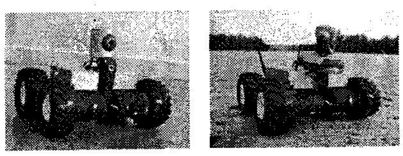


Fig. 6. *WiFiBot 4G* and *SC* models.

3.2 Basic Behaviours

A behaviour is a system that transforms the inputs from the sensors in an action pattern that will be used to carry out certain task. In object-oriented programming, a behaviour can be seen as a class with two methods, *perceptive_schema()*, which implements the algorithm for the perception and stores the results, and the method *motor_schema()*, that uses the perception to calculate an array which specifies the output direction.

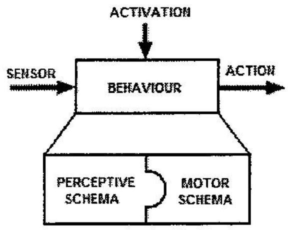


Fig. 7. Behaviour composed by the motor and perceptive schemas.

Then, in a behaviour-based system, there are several behaviours that, when active, depending on the sensory input, will return different action patterns that can be, for example, a linear and steering speeds or a direction angle (depending on the robot kinematics) in the case of robots navigation. This way, it is necessary to combine these different responses to build the global control action.

An architecture has been developed over the WiFiBot that allows the navigation while a team of robots maintain some pre-defined geometric shape. To do it, four basic behaviours have been implemented: *Go To Destination*, *Avoid Obstacles*, *Maintain Formation* and *Look For Reference*. These behaviours generate the control action that must be applied over the robot. This action consists on a linear speed and a turning speed. These basic behaviours are combined using fuzzy logic [12] to create two high-level behaviours: *Navigate*, that locates the destination and directs the robot towards it avoiding possible obstacles, and *Form*, which allows maintaining a constant position and orientation relative to another robot that is moving. This hierarchy is shown on fig. 8.

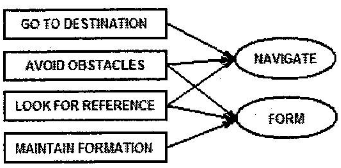


Fig. 8. Basic behaviours and its combination to generate high-level behaviours.

• ***Go To Destination***: This behaviour makes the robot move until it reaches a determined destination. It is composed of a perceptive schema and a motor schema as shown on fig. 9. The first one locates the target position for the robot (using the captured image as explained in the previous section). The motor schema calculates the speed arrow that brings the robot to the destination.

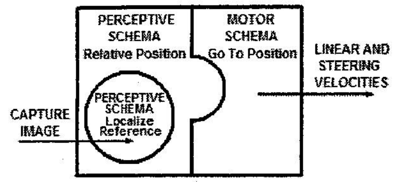


Fig. 9. Behaviour *Go To Destination*. Perceptive and motor schemas.

The perceptive schema *Relative Position* determinates the destination position making use of the position of the reference (mark) adding a relative position. This schema is used also by the behaviour *Maintain Formation*. In this case, it finds the position that the robot must have in the formation knowing the position of the leader, where the label will be located, as shown on fig 1. The position of leader respect follower is $[X_L, Y_L, T_L]^T$ and the one that must be found is $[X_d, Y_d, T_d]^T$:

$$\begin{pmatrix} X_d \\ Y_d \\ \theta_d \end{pmatrix} = \begin{pmatrix} X_L \\ Y_L \\ \theta_L \end{pmatrix} + \begin{pmatrix} \cos\theta_L & -\sin\theta_L & 0 \\ \sin\theta_L & \cos\theta_L & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} X_R \\ Y_R \\ \theta_R \end{pmatrix} \quad (9)$$

where $[X_R, Y_R, T_R]^T$ is the relative position respect the leader coordinate system to maintain the formation.

The motor schema *Go To Position* has been implemented using potential fields [13]. To achieve this goal, the space has been divided into three influence areas: *The ballistic zone*, where the robot is far from the destination. In this zone, the linear speed takes its maximum value and the steering one is proportional to the angle between the current robot direction and the direction to the destination. *The dead zone*, in which the robot is inside an error margin around the destination. In this zone, the speed is zero. Finally, *the controlled zone*, the medium zone where the advance speed is proportional to the distance that separates the robot from the destination, and the steering speed is proportional to the angle between the current direction of the robot and the direction to the destination.

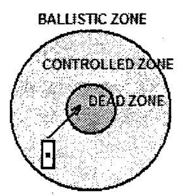


Fig. 10. Influence areas to calculate the velocity in the motor schema *Go To Position*.

Avoid Obstacles. It generates a repulsive effort when an obstacle is ahead the robot. It is also composed by two schemas, the perceptive one *Obstacle Position* and the motor one *Avoid*. The perceptive schema *Obstacle Position* calculates the position of a possible obstacle situated in front of the robot, taking into account the information of the two infrared sensors the *WiFiBot* robot has in its front. According to fig. 11, the distance to the obstacle can be calculated as:

$$
\begin{aligned}
X_0 &= (d_R + d_L)/2 \\
Y_0 &= 0 \\
\theta_0 &= \arctan((d_R + d_L)/D)
\end{aligned}
\qquad (10)
$$

The motor schema *Avoid* makes the robot turn when it finds an obstacle in front of it. The linear speed is null and the steering one is proportional to the distance to the obstacle, being null when the distance is over certain threshold value and maximum when the obstacle is situated at the minimum allowed distance.

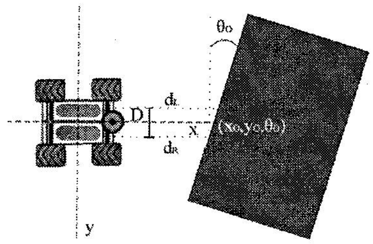


Fig. 11. Detection of the distance to a frontal obstacle.

• ***Look For Reference***. The robot moves around describing circles looking for the reference. Taking into account that the system does not provide communication between the robots and neither a common global reference system for all the robots, it is necessary to carry out first a research of the leader.

• ***Maintain Formation***. It allows the following of a mobile object keeping a relative position and orientation respect to it. It is composed by the same perceptive schema than the behaviour *Go To Destination* and a motor schema *Follow Reference* that is in charge of the activation of the behaviour when the reference is detected. The motor schema, knowing the position of the reference (leader), calculates the necessary control action (linear and steering velocities). To achieve it, the space is divided in several zones, as shown of fig. 12. In this case, it is not calculated the absolute linear speed but an increment over the previous one. This way, if the robot is in the dead zone, the current speed will not vary, and it will be updated (increased or decreased) linearly in the controlled zones, until the ballistic zones, where the increment will be the maximum allowed. The steering speed will be proportional to the orientation of the formation θ in the dead zone, and it will change linearly in the controlled zone until the ballistic zone, where it will take a value proportional to α. (Fig. 1). This way, in the ballistic zone, the robot tends to its correct position in the formation, and in the dead zone it takes the global direction of the formation.

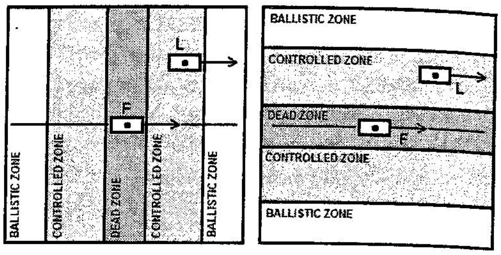


Fig. 12. Influence areas to calculate the linear velocity (left) and the steering velocity (right).

3.3. Coordination of behaviours

The previous basic behaviours must be combined in order to create a high level behaviour that has all the features of the basic ones. To do this, the basic behaviours will be added, assigning them a weight that depends of the current state of the robot. To obtain these weights, fuzzy logic is used. As an example, when the robot does not find any obstacle, the behaviour *Avoid Obstacles* must not be activated. However, if an obstacle is detected, the importance of the behaviour *Go To Destination* must decrease and *Avoid Obstacles* must increase its weight in the global action. Once the obstacle is avoided, the behaviour *Go To Destination* must recover its initial importance. Two high-level behaviours have been built:

• ***Navigate***. This behaviour, active in the leader, is the result of the combination of the basic behaviours *Go To Destination*, *Avoid Obstacles* and *Look For Reference*.

• ***Form***. This behaviour, active in the followers, is the result of the combination of the basic behaviours *Maintain Formation*, *Avoid Obstacles* and *Look For Reference*.

4. Experimental results

Several experiments have been carried out over a team of *WiFiBot* robots to test the performance of the high-level behaviours. Fig 13 shows the results obtained for the *Navigation* behaviour. Initially, the robot can not see the reference, and the purpose is to situate the robot at 0.8 m distance and 0° orientation to the reference. At first, the *Look For Reference* behaviour is the most important. It provides a constant steering speed to the robot. When the robot finds the reference (t = 40 s), the robot is directed to it stopping when it is situated in a threshold area around the objective. In t = 10 s, the robot finds an obstacle, what makes it to turn to avoid it (*Avoid Obstacles* behaviour). On fig. 14, the results for the *Form* behaviour are presented. The objective is that a robot follows the leader at 0.8 m distance and 0 rad orientation. As can be seen, the linear velocity increases to reach the same value as the leader.

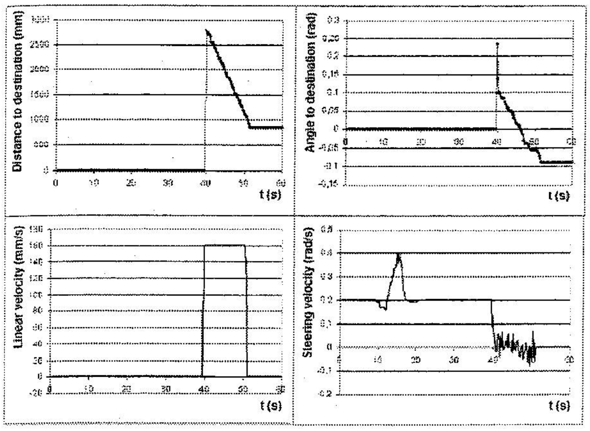


Fig. 13. Results for the *Navigate* behaviour.

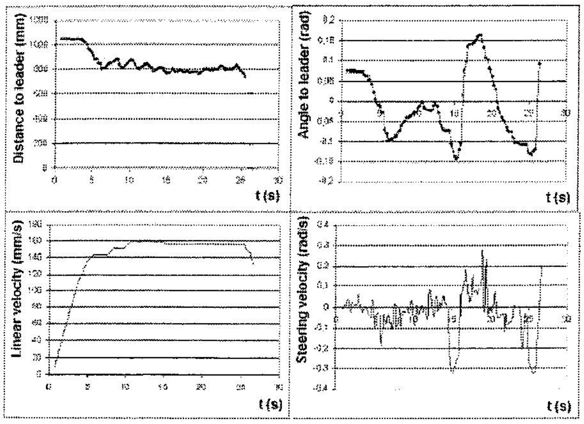


Fig. 14. Results for the *Form* behaviour.

5. Conclusion

A behaviour-based system for multi-robot formations control using computer vision has been presented. Through the combination of several basic behaviours using fuzzy logic, this system allows creating two high-level behaviours to control the velocities of the robots. The main goal of these two behaviours is to achieve that a team of robots navigate to reach a destination maintaining some predefined geometric shape between them and avoiding possible obstacles.

A robust and consistent method for estimating the position of the destination and the position of the leader robot using projective geometry has been developed and tested. Several experiments have been carried out to test the performance of the system over a team of *WiFiBot* robots.

Acknowledgements

This work has been supported by Ministerio Educación y Ciencia through DPI2004-07433-C02-01 'Herramientas de Teleoperación Colaborativa. Aplicación al Control Cooperativo de Robots' and the Project 'Localización y exploración mediante robots cooperativos e información multisensorial', founded by Generalitat Valenciana.

References

[1] Y.U. Cao, A.S. Fukunaga, A.B. Kahng & F. Meng, Cooperative mobile robotics: Antecedents and directions, *Autonomous Robots, 4*(1), 1997, 7-27.

[2] L. Iocchi, D. Nardi, M. Salerno, Reactivity and deliberation: a survey on multi-robot systems, *Lecture Notes in Computer Science, 2103*, 2001, 9-34.

[3] L.E. Parker, Current state of the art in distributed autonomous mobile robotics, *Proc. of the International Symposium on Distributed Autonomous Robotic systems*, Tokyo, 2000, 3-12.

[4] J.P. Desai, J. Ostrowski & V. Kumar, Controlling formations of multiple mobile robots, *Proc. of the IEEE International Conf. on Robotics and Automation*, Leuven (Belgium), 1998, 2864-2869.

[5] J. Spletzer, A.K. Das, R. Fierro, C.J. Taylor, V. Kumar & J.P. Ostrowski, Cooperative localization and control for multi-robot manipulation, *Proc. Conf. on Intelligent Robots and Systems*, 2001, 631-636.

[6] K.H. Tan & M.A. Lewis, Virtual structures for high-precision cooperative mobile robotic control, *Proc. Conf. on Intelligent Robots and Systems*, 1996, 132-139.

[7] J.C. Latombe, *Robot motion planning* (Kluwer Academic Publishers, Norwell, Mass, 1991).

[8] R. Arkin and T. Balch, Aura: principles and practice in review, *Journal of Experimental and Theoretical Artificial Intelligence, 9*(2), 1997, 175-189.

[9] T.D. Barfoot and C.M. Clark, Motion planning for formation of mobile robots, *Robotic and Autonomous Systems, 46*, 2004, 65-78.

[10] T. Balch and R. Arkin, Behavior-based formation control for multi-robot teams, *IEEE Transactions on Robotics and Automation, 14*(6), 1998, 926-938.

[11] M.A. Fischler and R.C. Bolles, Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography, *Communications of the ACM, 24*(6), 1981, 381-395.

[12] A. Saffiotti, The uses of fuzzy logic in autonomous robot navigation, *Soft Computing, 1*(4), 1997, 180-197.

[13] T. Balch and M. Hybinette, Social potentials for scalable multi-robot formations, *Proc. IEEE Int. Conf. on Robotics & Automation*, San Francisco, 2000, 73-80.