ELSEVIER

Pattern Recognition Letters 22 (2001) 1331–1335

Pattern Recognition Letters

www.elsevier.com/locate/patrec

Maintaining the relative positions and orientations of multiple robots using vision

António Paulino *, Jorge Batista, Helder Araújo

Department of Electrical Engineering, Institute of Systems and Robotics, University of Coimbra-Polo II, 3030 Coimbra, Portugal

Abstract

To keep several robots in geometric formation several issues have to be considered. This paper describes an experimental system for controlling multiple robots in geometric formation. This system has a mixed control structure, where both a central coordinator is used and also each robot has a high level of autonomy. The relative locations of the robots are estimated using vision. The central coordinator is used to decide the type of formation and to inform the robots about the formation type. Each robot is autonomous to make any decisions relative to the control of its position and velocity. Any of the robots can act as the central coordinator, but at any instant of time there is only one coordinator. The type of geometric formation can be dynamically changed and it depends on the overall goal of the system or on dynamic changes of the environment. To keep the robots in geometric formation their relative positions and orientations have to be known and they are estimated using vision.

Keywords: Mobile robots; Vision; Pose estimation

1. Introduction

Robot cooperation and coordination can be useful in different types of applications (Werger, 1998). In most cases the cooperation of multiple robots requires that their relative positions and orientations are known (Kurazume and Hirose, 1998). Some of the mobile robot cooperation applications involve keeping the robots in geometric formation. Examples of such applications range from fire fighting to military/security tasks and game playing (Balch and Arkin, 1998). Formation is a technique that is used to ensure that an area or region are efficiently covered by a set of robots, policemen, soldiers or players. For example in the MARTHA project (Alami et al., 1998) several results of co-operative mobile robotics were explored and each robot acts almost fully autonomously. Keeping the robots in formation may imply to keep constant their relative positions only or both the relative positions and orientations (Wang, 1991). The geometric pattern can also change dynamically as a function of the task (Yamaguchi, 1997; Lawton et al., 2000). In general several sources of error prevent the relative positions and orientations from being constant. To keep formation it is therefore essential to be able to measure deviations from the predefined pattern of relative positions and orientations. One important aspect of this problem is the control of the geometric formation (Sugihara and Suzuki, 1996; Pirjanian and Mataric, 2000). One of the main

* Corresponding author. Tel.: +351-39-796200; fax: +351-39-406672.

E-mail address: paulino@isr.uc.pt (A. Paulino).

0167-8655/01/$ - see front matter
PII: S0167-8655(01)00078-2

issues in this problem is the computation of the relative poses. If only the relative positions are considered several sensors can be used namely GPS, inertial or vision. If both relative positions and orientations are to be kept constant then, in general, only inertial and/or vision sensors can be used. However inertial sensors with a degree of accuracy allowing formation to be kept for extended periods of time are very expensive. A low cost alternative is the use of vision sensors. In the case of the system described in this paper we use vision as the sensor for the measurement of the robots poses and positions.

2. Formation control

The control of formation can be performed using centralized control or a distributed control structure. Both have advantages and disadvantages and in our case the control structure is mixed. There is a central coordinator that can be changed dynamically and each robot has a high degree of autonomy. In the event of a problem/fault of the central coordinator another robot will become the central coordinator. In each robot there is a central coordinator, hereinafter referred to as CC. At any instant of time only one CC is active. Before controlling the formation it is necessary to form the pattern. For that purpose the CC defines the relative position of each robot as well as the sequence of formation patterns. It receives the positions of each one of the robots and broadcasts all the positions to all the robots. In fact it is running in each robot, but at any instant of time only one of them acts as the coordinator. The low level control is decentralized (each robot has its own vector to control group formation).

During the initialization of the system, each robot defines its own world coordinate system. This world coordinate system is used by each robot to locate itself using odometry, and also to locate other robots of the group (using vision). Knowing the positions of the robots, and having the reference pattern defined by the CC, each robot can compute the reference position, and therefore the position error. Using this approach, each robot is completely independent of the others, which is useful in case of a breakdown in the communication system.

To control formation it is necessary to compute each robot pose error. Consider the follower robot i. Its pose $P = (x, y, \theta)$ as well as the leader position $P_L = (x_L, y_L, \theta_L)$, and its reference position $P_{ref_i} = (x_{ref_i}, y_{ref_i})$, relative to the formation leader, are known. The error vector is simply the difference $P_{err_i} = P_L + P_{ref_i} - P$. All these vectors are shown in Fig. 1 and are referenced in the robot world coordinate system. The control subsystem generates the velocity commands to cancel the error P_{err_i}. The translational and rotational errors, ΔS and $\Delta\alpha$, are used to obtain the corresponding translational and rotational velocities, V_T and V_α. These velocities are then controlled using a PID. To compute ΔS and $\Delta\alpha$, it is necessary to compute P_{err_i} in the robot coordinate frame, by means of a rotation of θ_i around the robot Z axis:

$$\begin{aligned} x^R_{err_i} &= \cos(\theta)x_{err_i} + \sin(\theta)y_{err_i}, \\ y^R_{err_i} &= -\sin(\theta)x_{err_i} + \cos(\theta)y_{err_i}. \end{aligned} \tag{1}$$

The translational and rotational errors can then be computed:

$$\begin{aligned} \Delta S &= \sqrt{x^R_{err_i}x^R_{err_i} + y^R_{err_i}y^R_{err_i}}, \\ \Delta\alpha &= \arctan\frac{y^R_{err_i}}{x^R_{err_i}}. \end{aligned} \tag{2}$$

A PID is used to control the translational and rotational velocities. These velocities are con-

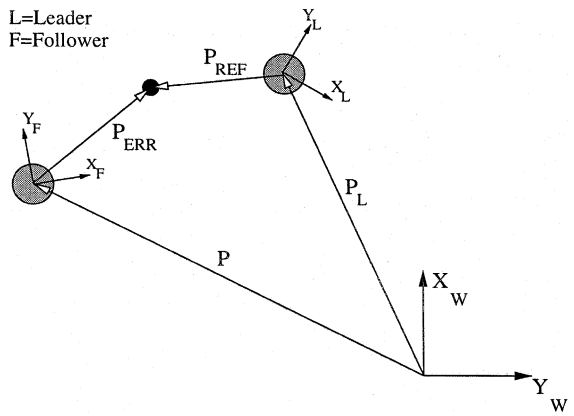

Fig. 1. Vectors used for formation maintenance.

verted into the robot model. In the case of this system, it is a two-wheel differential drive model. It has been proved (see Wang, 1991) that this definition of the error vector leads to a stable and controllable system. Therefore the robots always converge to the desired position in the formation. In the case of not having the leader position available, the system can use the (probably less accurate) position of other robots of the system, since robot i knows the reference position of the other robots.

3. Computing relative positions and orientations

Each robot estimates its pose relative to a visible neighbor. A planar pattern is used to enable robot identification and also pose estimation. For that purpose each robot carries a planar pattern similar to the pattern depicted in Fig. 2. To perform pose estimation and robot calibration the cameras have to be previously calibrated. They are calibrated using the method described in (Batista et al., 1999). Pose estimation based on the planar model requires the segmentation of the pattern image. The planar patterns that may be visible have to be located in each image. After image binarization, contour detection is performed in all the image. A list of closed contours in the image is thus estimated. The pattern contours, which are ellipses (the planar pattern is made up of black circles on a white background) are among the computed closed contours. The segmentation is based on least-squares ellipse fitting, using the algorithm described in (Fitzgibbon et al., 1999). The computation of the ellipses parameters is based on the solution of a generalized eigenvalues problem. After the ellipse fitting process, the estimated contours are projected into the image so that they are cross-checked against the image contours. This projection filters out estimated ellipses that do not correspond to the planar pattern. After localizing the pattern in the image, pose can be computed. In this application we assume only one degree of freedom in rotation, instead of three, and three degrees of freedom in the translation, which represents the vector from the camera coordinate frame and the reference coordinate frame. This is plausible, since each robot can only pan, and it moves on a XY plane. If we define the reference coordinate frame in such a way that the XY plane is parallel to the floor on which the robot moves, only one rotation angle needs to be considered. This rotation angle corresponds to the rotation around the Z axis. In this case, the rotation matrix between the camera coordinate frame and the reference frame is described by only one parameter. The image projection equation relating image coordinates with the corresponding 3D coordinates is

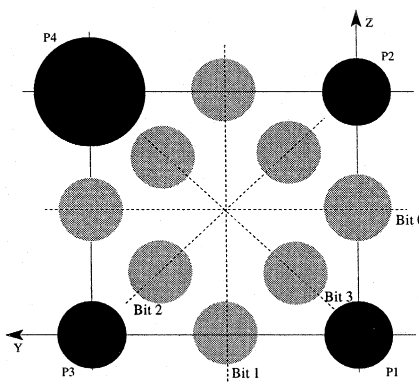

Fig. 2. Structure of the planar pattern used to locate and identify the robots.

$$[x_i \quad y_i \quad 1]^{\mathrm{T}} = \boldsymbol{H} \cdot [X \quad Y \quad Z \quad 1]^{\mathrm{T}}, \tag{3}$$

where $\boldsymbol{H}$ is a 3×4 matrix such that $\boldsymbol{H} = \lambda \cdot \boldsymbol{C} \cdot \boldsymbol{P} \cdot [\boldsymbol{R}|\boldsymbol{t}]$, where λ is a scale factor, $\boldsymbol{C}$ is the matrix of the camera intrinsic parameters, $\boldsymbol{P}$ is the projection matrix, $\boldsymbol{R}$ is the rotation matrix and $\boldsymbol{t}$ is the translation vector, in the camera coordinate frame. Assuming that the points of the planar pattern reference are located in the $X = 0$ plane, the projection equation can be rewritten as

$$\lambda \cdot [x_i \quad y_i \quad 1]^{\mathrm{T}} = \boldsymbol{G} \cdot [Y \quad Z \quad 1]^{\mathrm{T}}, \tag{4}$$

where matrix $\boldsymbol{G}$ is a 3×3 matrix. In our case (and since the camera is calibrated) this matrix is a function of the three translation parameters and the Z-rotation angle ϕ. Actually the vector of unknowns is $\boldsymbol{u} = [\sin\phi \quad \cos\phi \quad t_x \quad t_y \quad t_z]^{\mathrm{T}}$ Using a least-squares approach the five unknowns are computed. The values estimated for the translation vector have good accuracy and stability. However the values estimated for ϕ are both inaccurate and unstable. In fact, when ϕ is close to zero, its estimates are much more affected by noise than for any other range of values. Since these errors significantly affect the performance of the formation control algorithms, only the values computed for translation are used. The translation vector is not affected by the angle between the axes, and therefore its estimates can be used to get a better estimate of the angle. Since the reference pattern is known, the knowledge of the translation vector allows the computation of the coordinates of the reference pattern points in the camera coordinate frame. Considering two points of the reference pattern and their camera coordinates $\boldsymbol{P}_{1C}$ and $\boldsymbol{P}_{2C}$, ϕ can be computed by

$$\phi = \arcsin \frac{P_{2C_z} - P_{1C_z}}{|\boldsymbol{P}_{2C} - \boldsymbol{P}_{1C}|}. \tag{5}$$

The estimates for ϕ obtained from the translation estimates are more stable and robust than those directly estimated from the image projection equation.

In the case of the formation control described in Section 2, we need the measurements relative to the robot world, instead of the 3D position and orientation relative to the camera, given by the pose estimation method, which means the image measurements must be translated to this reference frame. Since we know the robot position (through odometry readings), we must first compute the estimated pose in robot coordinates. In our system, each robot has three cameras, covering a range of about two hundred degrees. Each camera can be described with a pose $P_{ci} = (x_{ci}, y_{ci}, \theta_{ci})$, relative to the robot. Knowing that the estimated pose $P_e = (x_e, y_e, \theta_e)$ is obtained with camera i, the same estimated pose is, in robot coordinates:

$$\begin{aligned} x_e^R &= \cos(\theta_{ci})x_e + \sin(\theta_{ci})y_e + x_{ci}, \\ y_e^R &= -\sin(\theta_{ci})x_e + \cos(\theta_{ci})y_e + y_{ci}, \\ \theta_e^R &= \theta_e + \theta_{ci}. \end{aligned} \tag{6}$$

A similar coordinate transformation is then used to bring P_e^R to the world frame, giving us the P_L described in Section 2.

4. Experimental results

The whole architecture was implemented and tested, in a robot simulator and in the field. This section presents some results of these tests performed with Nomadic Superscout robots. These results describe the performance of an integrated system that includes the estimation of the robot poses (and therefore of the reference vectors defining formation) as well as the correction of the errors in the vectors describing the relative positions and orientations of the robots. Fig. 3 shows the laboratory and the Superscouts. In these tests, the reference pattern was located on the leader, and both followers compute the leader's pose, relative to their own world, as considered in Section 2. In this way, the only constraint is that all the followers are started with their X axis parallel to each other. Fig. 4 shows two of the tests performed with the system.

The first test was done using only one robot following the leader, and having a reference position of $(-140, 10)$, in the leader coordinate system. As it can be seen, the follower can track the leader with good accuracy, and maintaining an almost null error. In the second test, another robot was introduced. Its reference formation vector, in the leader reference frame, $(-140, 0)$. Again the error is near zero. However, in these tests, the panoramic camera mentioned in Section 3 was not used. Instead, a 2/3″ CCD camera, with a 6.5 mm focal length was used, yielding us a field of view of about 60°.

5. Conclusions

In this paper an system for controlling geometric robot formation was described. Robot

Fig. 3. Top view of the lab and the robots.

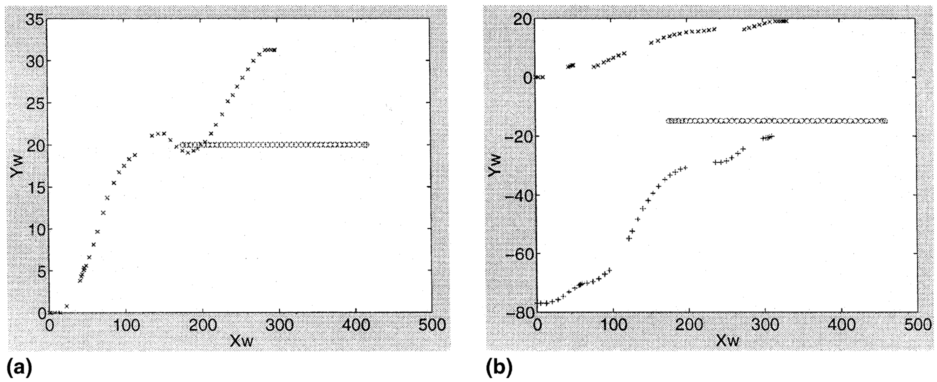

Fig. 4. Results with Superscouts: (a) *X* and *Y* positions (in the room coordinate frame) of the leader and follower robots; (b) *X* and *Y* positions (in the room coordinate frame) of the leader and the two follower robots.

positions and orientations are estimated using vision. The system described has the ability to change dynamically the pattern of formation. During the course of operation the set of robots can dynamically adjust the pattern of formation depending on environmental conditions or functional goals. Three main problems were dealt with: control of formation, dynamic change of formation while avoiding self-collision and robot location using vision. Current extensions of this work include the use of panoramic cameras to increase the operating range of the robots.

Acknowledgements

The authors gratefully acknowledge the support of projects PRAXIS/2/2.1/TPAR/2074/95, and PRAXIS/P/EEI/10252/1998 funded by the Portuguese Foundation for Science and Technology. António Paulino was supported by project PRAXIS/P/EEI/10252/1998.

References

Alami, R., Fleury, S., Herrb, M., Ingrand, F., Robert, F., 1998. Multi-robot cooperation in the MARTHA project. IEEE Robot. Automat. Mag., 36–47.

Balch, T., Arkin, R.C., 1998. Behavior-based formation control for multi-robot teams. IEEE Trans. Robot. Automat. 14 (6).

Batista, J., Araújo, H., Almeida, A.T., 1999. Iterative multistep explicit camera calibration. IEEE Trans. Robot. Automat. 15 (5).

Fitzgibbon, A., Pilu, M., Fisher, R., 1999. Direct least squares fitting of ellipses. In: IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 21, N. 5, May 1999; 13, N. 4, April 1991.

Kurazume, R., Hirose, S., 1998. Study on cooperative positioning system-optimum moving strategies for CPS-III. In: Proc. ICRA'98 – IEEE Internat. Conf. on Robotics and Automation, May 16–20, Leuven, Belgium, pp. 2896–2903.

Lawton, J.R., Young, B.J., Beard, R.W., 2000. A decentralized approach to elementary formation manoeevers. In: Proc. 2000 IEEE Internat. Conf. on Robotics and Automation, San Francisco, CA, April 2000, pp. 2728–2733.

Pirjanian, P., Mataric, M., 2000. Multi-robot target acquisition using multiple objective behavior coordination. In: Proc. 2000 IEEE Internat. Conf. on Robotics and Automation, San Francisco, CA, April 2000, pp. 2696–2702.

Sugihara, K., Suzuki, I., 1996. Distributed algorithms for formation of geometric patterns with many mobile robots. J. Robot. Systems 13 (3), 127–139.

Wang, P.K.C., 1991. Navigation strategies for multiple autonomous mobile robots moving in formation. J. Robot. Systems 8 (2), 177–195.

Werger, B.B., 1998. Principles of minimal control for comprehensive team behavior. In: Proc. of the ICRA'98–IEEE Internat. Conf. on Robotics and Automation, May 16–20, Leuven, Belgium, pp. 3504–3509.

Yamaguchi, H., 1997. Adaptive formation control for distributed autonomous mobile robot groups. In: Proc. 1997 IEEE Internat. Conf. on Robotics and Automation, Albuquerque, New Mexico, April 1997, pp. 2300–2305.