On-Board Vision Using Visual-Servoing for RoboCup F-180 League Mobile Robots

Paul Lee, Tim Dean, Andrew Yap, Dariusz Walter,
Les Kitchen, and Nick Barnes

Department of Computer Science and Software Engineering
The University of Melbourne
Melbourne, VIC 3010, Australia
http://www.cs.mu.oz.au/robocup/2003/F180/
publications/2002/localvis/index.php

Abstract. In the RoboCup F-180 league competition, vision is predominantly provided by an overhead camera which relays a global view of the field. There are inherent disadvantages in utilising this system, particularly the delays associated with the capture, transmission and processing of vision data. To minimise these delays and to equip the robots with greater autonomy, visual servoing on-board the individual robots is proposed. This paper presents evaluation of two visual servoing methods for mobile robots: position-based and image-based servoing. Traditional implementations of image-based servoing have relied on partial pose estimation, negating much of the advantage gained from using this method. This paper will present an alternative implementation of image-based servoing for approaching objects on the ground plane, which disposes of the pose estimation step and fully relies only on image features. To evaluate the suitability of both visual servoing methods to F-180, the task of docking with the ball is used as a basis of the investigation.

1 Introduction

The implementation of vision as a form of feedback for robotic tasks is a major field of research dating back to the 1970s, where much of the early investigation concentrated on pattern recognition problems [6]. Owing to the reduction in hardware costs and the increase in computing power, the focus of vision research has turned to the introduction of visual data into the control loop of a robot. Using visual data within the control loop has been termed *visual servoing*.

The classical approaches to visual servoing are position-based servoing and image-based servoing. In position-based servoing the world pose of the target is estimated from the image, generally based on a geometric model of the object, and a position-based control signal is generated accordingly. In image-based servoing, the motion is controlled directly based on information from the image plane, with the control error signal being the difference between a desired feature vector, and the current feature vector. However, there are problems with image-based servoing such that large undesired motions of the robot can occur, and

D. Polani et al. (Eds.): RoboCup 2003, LNAI 3020, pp. 422–433, 2004.

the object may move out of camera view [10, 12]. To deal with these problems, recent work has examined $2\frac{1}{2}$D visual servoing, where the camera displacement between current and desired positions is estimated in 3D coordinates without the need for a 3D model of the target [8]. Other work emphasises robot path planning in image space to avoid the problems of loosing the object from the field of view [10, 12].

In this paper, we examine visual servoing for on-board control in the RoboCup F-180 League. Currently, vision for most F-180 teams is provided by an overhead camera which relays a global view of the field and all gameplay information back to a host computer. The disadvantages of a global vision system are the delays inherent in the capture, transmission and processing of the image by the host, which then issues commands to the robot. By employing vision on-board each individual robot, and using visual servoing to produce a much 'tighter' closed-loop control, decisions can be made by the robot without the intervention of the host.

The possibility of integrating both global and on-board vision provides the advantage of being able to select whichever form of vision is most appropriate for a particular task. For reflex actions such as aiming and shooting at goal, on-board vision would most likely be appropriate. Higher-level actions such as team strategies would be handled by global vision.

In F-180, the environment is well-known so a model of the object is available, facilitating position-based servoing. Further, in dealing with specific tasks such as chasing the ball, the problem is well posed for image-based servoing. We present the algorithms for both these approaches, and demonstrate through real robotic experiments using the University of Melbourne entries to the F-180 League as the platform for experimentation that both approaches are suitable for the F-180 league.

2 Theory

2.1 Position-Based Servoing

In position-based control, the task of positioning the robot is based on

1. extraction of image features during iteration of the control loop, and
2. evaluation of an estimate of the target pose with respect to the camera.

An error signal is determined from the difference between the current and desired pose. This error signal acts as an input to the system control law, as shown in Figure 1. Corke [3] states that the advantage of position-based control is that it neatly separates the computation of the feedback signal from the estimation problems involved in computing the pose from visual data. Corke, Hager and Hutchinson [7] contend that, in general, image-based control is preferable to position-based control due to factors such as positioning accuracy of the system being less sensitive to camera calibration, and the computational advantages which can be gained from a reduced number of transforms. However, Martinet

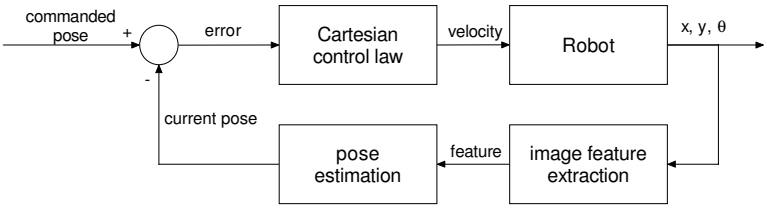

Fig. 1. Position-Based Control System

and Gallice [9] demonstrated that using non-linear state feedback, 3D visual features can still be incorporated into the control loop and achieve performance comparable to image-based servoing.

2.2 Image-Based Servoing

Image-based control consists of specifying the positioning task directly from the image without an estimation of the pose of the target. Feedback is purely from the image plane, and the error signal is computed as the difference between the desired feature vector $\mathbf{f}_\mathrm{d}$ and the current feature vector $\mathbf{f}$. A *feature vector* is a set of visual features such as the coordinates of vertices or the areas of the faces of an object. Elements of the task are therefore specified in the image space rather than the world space. i.e. in pixels rather than Cartesian coordinates.

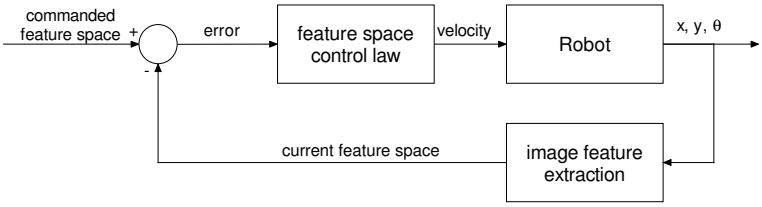

Fig. 2. Image-Based Control System

It is ideal in image-based control to reduce an appropriate error function e such that when the desired position is achieved, e is 0. While the error function is defined in the image parameter space, the input to the robot is in the task space, as shown in Figure 2. Therefore, in order to relate changes in image features to changes in the position of the robot, the concept of an image Jacobian is introduced [3, 4, 7, 11]. The following relationship is then given,

$$\dot{\mathbf{f}} = \mathbf{J}\dot{\mathbf{p}} \tag{1}$$

where $\mathbf{p} = [x, y, z]^T$ is a point in the body frame and $\dot{\mathbf{p}}$ the velocities of this point. As shown by Sharma, Sutanto and Varma [11], the Jacobian is evaluated from

$$\mathbf{J} = \frac{\partial \mathbf{f}}{\partial \mathbf{p}} = \begin{bmatrix} \frac{\partial f_1}{\partial p_1} & \frac{\partial f_1}{\partial p_2} & \cdots & \frac{\partial f_1}{\partial p_n} \\ \frac{\partial f_2}{\partial p_1} & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ \frac{\partial f_m}{\partial p_1} & \cdots & \cdots & \frac{\partial f_m}{\partial p_n} \end{bmatrix} \tag{2}$$

Assuming that the image Jacobian is square and non-singular, a simple control law can be determined as

$$\dot{\mathbf{p}} = \mathbf{k}\mathbf{J}^{-1} \left(\mathbf{f}_{\mathrm{d}} - \mathbf{f}(t)\right) \tag{3}$$

where $\mathbf{k}$ is a diagonal gain matrix which will implement simple proportional control.

Image-based control possesses underlying weaknesses since visual data is interpreted using a constantly refined Jacobian matrix as part of the control loop. Chaumette and Malis [1, 2] exposed the possibility of a local minimum being reached and the image Jacobian being singular during servoing. In order to avoid these problems, $2\frac{1}{2}$D visual servoing combines visual features obtained directly from the image with position-based features [8].

3 Implementation

3.1 Position-Based Servoing

Since an image is two-dimensional, it is difficult to extract three-dimensional (3D) Cartesian coordinates, because depth is unknown. However, the 3D coordinates are required to properly implement the control system. In order to calculate the depth of the centroid of an object, an assumption was made that the object will always be on the ground and of known size, which is reasonable to assume for the ball. Since the camera height and tilt angle were known, the position of any image objects was determined by triangulation. Once the position of the object was determined, a simple proportional and derivative controller was used regulate the motion of the robot to achieve the docking position.

3.2 Image-Based Servoing

For the simple task of docking the robot with a ball, the degrees of freedom considered were translation forwards and backwards, and rotation about the current position of the robot. Translation sideways was deemed redundant, since this can be achieved through a combination of forward motion and rotation.

The pinhole camera model was selected for the perspective projection, and the feature vector $\mathbf{f} = [u, v]^T$ was utilised, where u and v represent the image coordinates of the centroid of the object. Using the coordinate system in Figure 3, the velocity of a point $\mathbf{p}$ was expressed relative to the camera frame as,

$$\dot{x} = z\omega \tag{4}$$

$$\dot{y} = 0 \tag{5}$$

$$\dot{z} = -x\omega + v_x \tag{6}$$

where v_x and ω were the chosen degrees of freedom shown in Figure 4. Note that the camera used for on-board vision had non-unit aspect ratio and was oriented on its side to achieve the greatest possible depth of vision. Thus to reflect the reorientation of the camera, the normal convention of having u and v representing the horizontal and vertical coordinates was changed (see equation 7). As well as viewing the ball when docked with the robot, it was considered more important to be able to perceive objects at a distance, rather than to have a wider viewing range close to the robot. This is reflected in the image coordinate system.

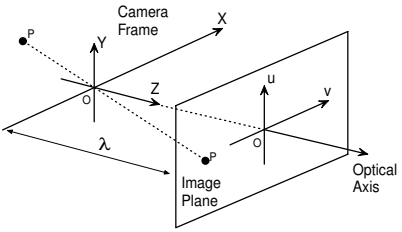

Fig. 3. Perspective projection of a pinhole camera

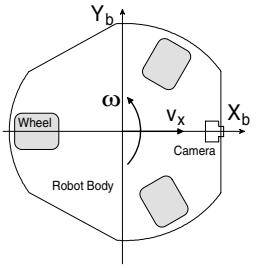

Fig. 4. Top view of configuration of omni-driven Kanga robot

The perspective projections

$$u = \lambda \frac{y}{z} \qquad \text{and} \qquad v = \lambda \frac{x}{z} \tag{7}$$

were used to express the velocities in (4) in terms of the feature parameters

$$\dot{x} = z\omega \tag{8}$$

$$\dot{y} = 0 \tag{9}$$

$$\dot{z} = -\frac{vz}{\lambda}\omega + v_x \tag{10}$$

Using these relationships the differential change in the image features was

$$\dot{v} = \lambda \frac{z\dot{x} - x\dot{z}}{z^2} \tag{11}$$

$$= -\frac{v}{z} v_x + \left(\frac{\lambda^2 + v^2}{\lambda} \right) \omega \tag{12}$$

$$\dot{u} = \lambda \frac{z\dot{y} - y\dot{z}}{z^2} \tag{13}$$

$$= \frac{uv}{\lambda} \omega - \frac{u}{z} v_x \tag{14}$$

Equations (12) and (14) can be written in matrix form as,

$$\begin{bmatrix} \dot{u} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} -u/z & uv/\lambda \\ -v/z & (\lambda^2 + v^2)/\lambda \end{bmatrix} \begin{bmatrix} v_x \\ \omega \end{bmatrix} = \mathbf{J}\mathbf{v} \tag{15}$$

The relationship between robot velocities and changes in the image is therefore

$$\mathbf{v} = \mathbf{J}^{-1}\dot{\mathbf{f}} = \begin{bmatrix} -z(\lambda^2 + v^2)/\lambda^2 u & vz/\lambda^2 \\ -v/u\lambda & 1/\lambda \end{bmatrix} \dot{\mathbf{f}} \tag{16}$$

A simple proportional controller was used to regulate the motion of the robot for image-based servoing. The aim of this investigation was not to examine the performance of the control system, but rather to focus on the characteristics of the visual-servoing methods. The simplest control system was therefore chosen.

In order to find the inverse Jacobian, $\mathbf{J}$ must be square and non singular, that is, the determinant must be non-zero.

$$|\mathbf{J}| = -u\lambda/z - u \cdot v^2/(z\lambda) + u \cdot v^2/(z\lambda)$$
$$= -u\lambda/z \tag{17}$$

According to Equation (17), the determinant can be zero only if either u is zero, or z is infinite. However, in practice these occurrences were defined as a no-ball case at the image-processing stage and handled prior to the control system. Thus, the Jacobian could be considered as always non-singular. Furthermore, in our experiments the underlying error function was monotonic and no local minima occurred in the control loop.

The derivation given for the image Jacobian considered the camera axis horizontal to the ground. However, for this implementation, the camera was tilted to enable the ball to be seen when at the base of the robot. The main effect of the tilt angle is to introduce a constant scaling factor into the robot control velocities. This scaling factor can be absorbed by the gain factor ($\mathbf{k}$) in the control law. It was found through experimentation that the appropriate selection of the controller gains results in the derived control law being effective even with camera tilt. The non-tilt derivation can therefore be used in the general case.

As discussed by Corke [3], many image-based models still require the depth of the target in the formulation of the image Jacobian. This results in a partial pose estimation, which is the basis of the position-based model. By utilising the image size of the target as an indicator of the relative distance from the

camera, Zhang and Ostrowski [12] were able to remove the dependence of the image Jacobian on the depth of the target. However, for the implementation of image-based control used in this investigation, image size was not considered an accurate representation of the depth because the entire object might not be recognised. This was a major problem encountered, as specular reflections and self-shadowing under the F-180 competition overhead lighting conditions caused variable recognition of the ball. In F-180 most dominant teams rely only on overhead cameras, and the lighting design is more suitable for overhead applications. However, the centroid height was determined to be a more robust ball feature when the ball is close, as partial occlusion will only change the distance estimate slightly. Although it does cause larger errors when the ball is far away, the control behaviour will not be significant different compared with when it is close by. Depth was therefore expressed as a function of the chosen image features and the geometry of the camera placement, as shown in Figure 5.

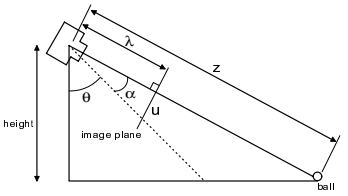

Fig. 5. Camera Placement

Using the geometry of the camera placement shown in Figure 5, the depth z and angle α were evaluated to be

$$z = \frac{h}{\sin\theta\cos\alpha + \cos\theta\sin\alpha} \tag{18}$$

$$\alpha = \arctan\left(\frac{u}{\lambda}\right) \tag{19}$$

Since the angle of elevation of the camera is known, both $\sin\theta$ and $\cos\theta$ were evaluated as constants. Furthermore, using (19) and Pythagoras' Theorem, $\cos\alpha$ and $\sin\alpha$ can be evaluated, allowing z to be determined.

4 Experimental Procedure

Two forms of experiments were designed to analyse the behaviour of each method of visual servoing: docking with a static ball, and docking with a moving ball. Both experiments performed docking with an orange golf ball, as used in RoboCup competition. The purpose of these tests was to observe the effects of system parameters such as frame rate, computation time and calibration.

In the static experiment the ball was placed at three locations in front of the robot: to the left, in the center, and to the right of the robot. The ball

was also placed at differing distances from the robot: near, middle and far away from the robot respectively. Figures 6(a) and (b) illustrate where the placement locations were placed relative to the robot. The ruler seen in these images is 1m in length. For the moving target tests, the ball was released from a ramp of fixed height and rolled towards the robot. The ramp allowed reasonable repeatability of tests with a consistent velocity. The robot was required to track varying ball trajectories at 90°, 45°, and almost parallel to the robot's line of vision.

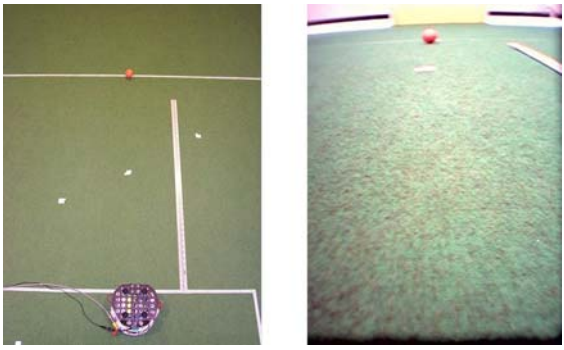

Fig. 6. Placement of the ball and test locations relative to the robot, as seen from **(a)** an overhead view and **(b)** the robot camera

5 Results and Discussion

Two sets of results for each experiment were collected for analysis. The results were of the overhead view of the test and of the on-board camera view. The overhead camera provided a world-space view of the path taken by the ball and the robot as docking occurred. The overhead view was also used to track changes in velocities as the robot approached the docking position. This was achieved by knowing the sampling rate used for the camera. The on-board view was used to observe the changes in ball position within the image frame. This data provided a better understanding of the servoing behaviour of the robot as it attempted to reach the commanded position. Note that the overhead camera was used *only* to record the motion of the robot for evaluation, *not* for servoing, which was done by on-board vision.

A representative sample of the results is given in Figures 7 to 9. Figure 7 presents the tests as seen by the on-board camera. Figures 8 and 9 present the global view of the tests seen by the overhead camera. The overhead plots contain markers at one-second intervals, while the local camera has markers at half-second intervals. The markers for image-based control are depicted by the "$\circ$", and for position-based control by the "$*$" marker. Complete results for all tests are given in [5].

From the results obtained there was no discernible difference in the performance of the two methods. Both servoing implementations were able to track

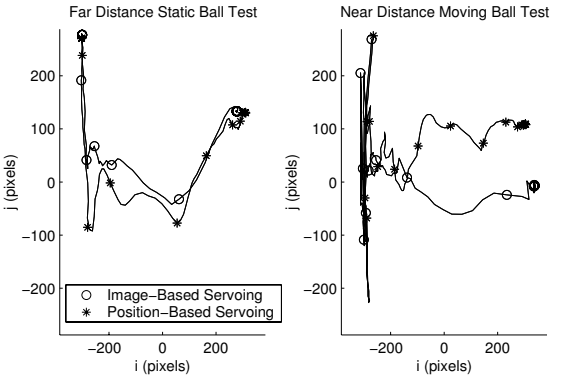

Fig. 7. Local camera view of two tests

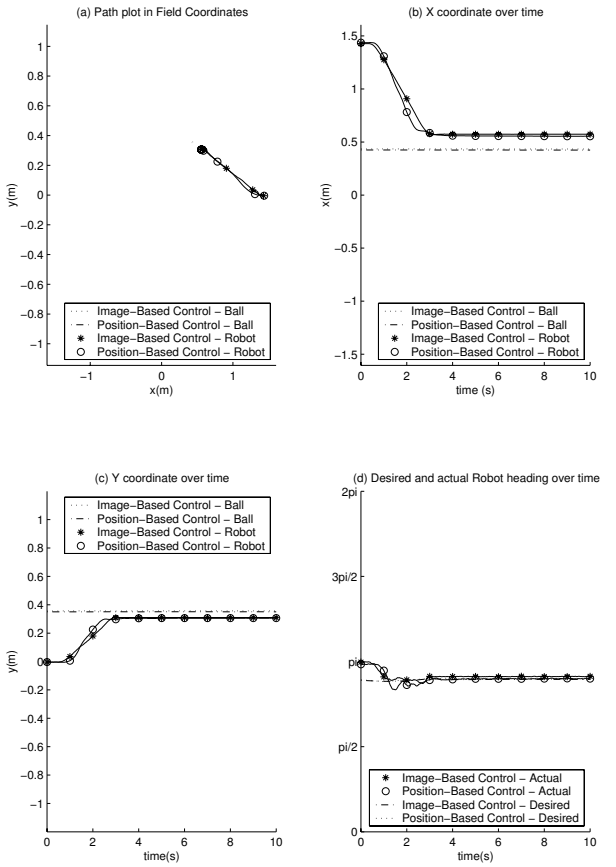

Fig. 8. Overhead camera view of far distance static ball test

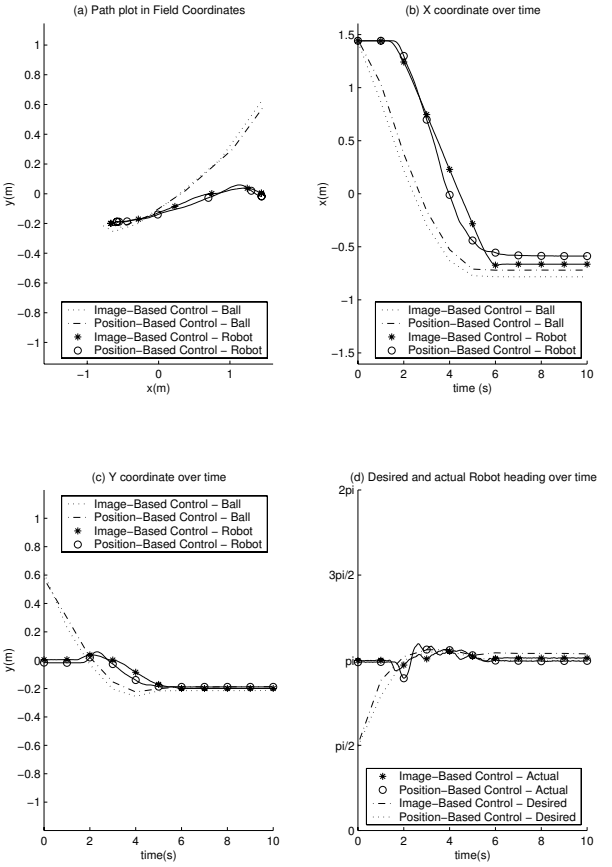

Fig. 9. Overhead camera view of near distance moving ball test

the ball reliably, and were able to successfully approach the docking position. Neither method was able to achieve the exact docking position due to the inability of the robot to move when given low-velocity commands. This was a potential control system problem which could be solved through the implementation of more advanced controllers. However, the simple controllers used for the experiments still demonstrated that visual servoing using either method can be satisfactorily achieved. Furthermore, during operation in the actual F-180 competition it would not be desirable to stop next to the ball, but rather to move to the ball at the highest speed possible in order for it to be picked up on a roller.

It was noticed that for the image-based method that the robot travelled at constant velocities to the docking position, whilst the velocities for the position-based method had a ramping effect. This is attributed to the different controllers used for each implementation. A proportional controller was used for the image-based method and accounted for the constant or linear velocities observed. For the position-based implementation, a derivative controller was added, introduc-

ing the ramping effect. The ramping effect was due to the predictive nature of a derivative controller, evidenced by the reduction in the commanded speed as the robot approached the docking position.

The position-based method was a much more intuitive means of visual servoing than the image-based method. Position-based servoing is essentially an extension of image-based servoing by the addition of a pose estimation step. Both methods required the extraction of image features in the feedback loop. The derivation of the Jacobian proved difficult to analyse for errors due to the coupling involved in the evaluation of the velocities. The position-based servoing was therefore a more straightforward method to conceptualise and implement.

In terms of computational costs, image-based servoing was much cheaper than position-based servoing. The removal of the transformation from image space to Cartesian world space allowed computation involving only addition, subtraction, multiplication, and division operations. The additional cost of using tan and arctan operations made position-based control much more computationally expensive. It needs to be considered, though, that when the complexity of the task increases, the computational cost of the image-based method would increase in proportion to the size of the Jacobian matrix. Image-based servoing could therefore potentially be more expensive than position-based methods for complex tasks.

In an application such as visual servoing calibration effects are an important issue to be considered. It was observed that for position-based servoing, external and internal calibration issues such as camera positioning and angle of field-of-view will cause errors in position estimation. However in image-based servoing, the main effect of calibration was from determination of the focal length, an integral step in the development of the image Jacobian. To minimise the erroneous influences of calibration and to achieve reliable servoing, it was therefore important for both servoing methods that the calibration be conducted offline.

6 Conclusion

In this paper, it has been shown that it is possible for visual servoing to be performed on board the F-180 robots under RoboCup conditions. The two classical methods of position-based and image-based servoing both satisfactorily achieved the task of docking a robot with static and moving targets. In conducting this investigation it was shown that both methods had advantages and disadvantages over each other and neither proved to be a superior implementation. As such, both are equally suitable to be used in the F-180 League, and the decision on using either method would be based on which features of position-based and image-based servoing are more desirable.

References

1. F Chaumette and E. Malis. $2\frac{1}{2}$D visual servoing: A possible solution to improve image-based and position-based visual servoing. In *Proc. IEEE Int. Conf. on Robotics and Automation*, volume 1, pages 630–635, 2002.

2. François Chaumette. Potential problems of stability and convergence in image-based and position-based visual servoing. In *The Confluence of Vision and Control*, number 237 in LNCIS Series, pages 66–78. Springer-Verlag, 1998.
3. P I Corke. *Visual Control of Robots: High Performance Visual Servoing*. Research Studies Press Ltd., Great Britain, 1996.
4. P I Corke and S A Hutchinson. A new hybrid image-based visual servo control scheme. In *Proc. IEEE Conf. on Decision and Control*, volume 3, pages 2521–2526, 2000.
5. T Dean, P Lee, and A Yap. Mechatronics research project: Local vision for small league robots – technical report. Technical report, The University of Melbourne, 2002. Available: `http://www.cs.mu.oz.au/robocup/2003/F180/publications/2002/localvis/Documents/Technical%20Report%202.15.pdf`.
6. B Espiau, F Chaumette, and P Rives. A new approach to visual servoing robotics. *IEEE Transactions on Robotics and Automation*, 8(3):313–326, 1992.
7. S Hutchinson, G Hager, and P I Corke. A tutorial on visual servo control. Technical report, CSIRO, Australia, 1996. Available: `http://www.cat.csiro.au/cmst/staff/pic/vservo.htm`.
8. E Malis, F Chaumette, and S Boudet. $2\frac{1}{2}$D visual servoing. *IEEE Transactions on Robotics and Automation*, 15(2):238–250, Apr. 1999.
9. P Martinet and J Gallice. Position based visual servoing using a non-linear approach. In *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, volume 1, pages 531–536, 1999.
10. Y Mezouar and F Chaumette. Path planning in image space for robust visual servoing. In *Proc. Int. Conf. on Robotics and Automation*, pages 2759–2764, 2000.
11. H Sutanto, V Varma, and R Sharma. Image based autodocking without calibration. In *Proc. IEEE Int. Conf. on Robotics and Automation*, volume 2, pages 974–979, 1997.
12. H Zhang and J P Ostrowski. Visual motion planning for mobile robots. *IEEE Transactions on Robotics and Automation*, 18(2):199–208, 2002.