Real-Time Vision Processing for a Soccer Playing Mobile Robot

Gordon Cheng and Alexander Zelinsky

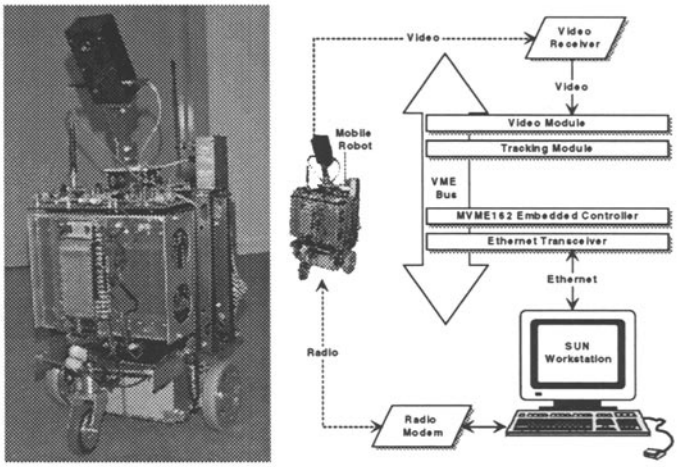
Department of Systems Engineering
Research School of Information Sciences and Engineering
The Australian National University
Canberra, ACT 0200. Australia
http://wwwsyseng.anu.edu.au/rsl/

Abstract. This paper describes vision-based behaviours for an Autonomous Mobile Robot. These behaviours form a set of primitives that are used in the development of basic soccer skills. These skills will eventually be utilised in an effect to tackle the challenge that has been put forward by the "The Robot World Cup" initiative. The focus of the discussion will be on the vision processing associated with these behaviours. Experimental results and analysis of the visual processing techniques are also presented.

1 Introduction

In this paper we will discuss the basic soccer skills we have developed for a mobile robot. A self-contained autonomous mobile robot in a physical game of soccer provides an excellent challenge for robotics research. Under the initiative of the "The Robot World Cup" (RoboCup) [Kitano et al., 1997] the challenge was to construct robots to play a game of soccer. The system is our preparation for the up and coming RoboCup events (e.g. RoboCup'98). Our current focus is on the primary sensing capabilities of the robot with emphasis on high and robust performances for the overall system. We believe providing a robot with specialised skills must come first, and must not be ignored before moving onto the design of higher level skills. We have commenced work on the construction of visual based behaviours for this challenge. We have committed to the development of localised vision sensing for our robot.

In order for a robot to play soccer it must be able to react to its changing external environment quickly. It must also be able to navigate freely on the playing field while avoiding any obstacles in its way. A Behaviour-based [Brooks, 1986] approach for the construction of our system have been chosen. Our system are built from individual competence modules; which involve a strong coupling of the robot's sensors to its actuators. The end-effect of the collective interaction of the individual modules with the external environment produces a competent and coherent control system for the entire robot.



a) b)

Fig. 1. System Configuration: a) Yamabico Mobile Robot b) System Overview

2 Our System

The configuration of our experimental system is shown in Figure 1 consists of three components, an Off-board vision processing unit, a Radio communication unit, and a Mobile Robot. The vision processor is a Fujitsu MEP vision system. It comes with two processing modules, a video module and a tracking module. The system is designed to support five tracking module simultaneously. These modules are connected via a VME-bus backplane to a Motorola MVME-162 Embedded Controller running the VxWorks operating system. The VxWorks operating system handles all program executions. A tracking module can track up to 100 templates in each video frame in a 30 Hz video stream. Currently we are using one tracking module. The video module provides two selectable input channels and one output channel. All signals are NTSC video format. A NTSC video receiver is connected to one of the input channels on the video module. This receiver is used to accept video signals from the mobile robot. The communications system is a SUN-workstation (running Solaris OS) with a radio modem attached. The communication system manages all network traffic between the vision system and our mobile robot(s). For example, once a image frame is processed, a command from the vision system is send back to the robot, guiding it on its way.

The Yamabico mobile robot shown in Figure 1a [Yuta et al., 1991]. It has a multi-processor based architecture that houses a number of processor modules. All of these modules communicate through the Yamabico-bus, using a Dual-Port-Memory mechanism. The robot has a MC68000-CPU master module, running the Morsa OS [Yuta et al., 1991]. A T-805 locomotion module, that provides all of the motor feedback and control of the robot [Iida and Yuta, 1991]. An ultrasonic module is also provided on the robot, it is not used in our experiments.

In addition, a radio modem, a small size CCD camera and a video transmitter has been included in our system. The modem is used to communicate with the communication system. The video transmitter and camera provide video input to the vision system.

2.1 Information flow

The basic information flow of the system is the robot transmits video signals taken by the on-board CCD camera. A vision system connected to a video receiver receives the signals for processing and then sends the results to a SUN workstation. The workstation transmits the results back to the robot via the radio modem. An overview of this organisation is shown in Figure 1b.

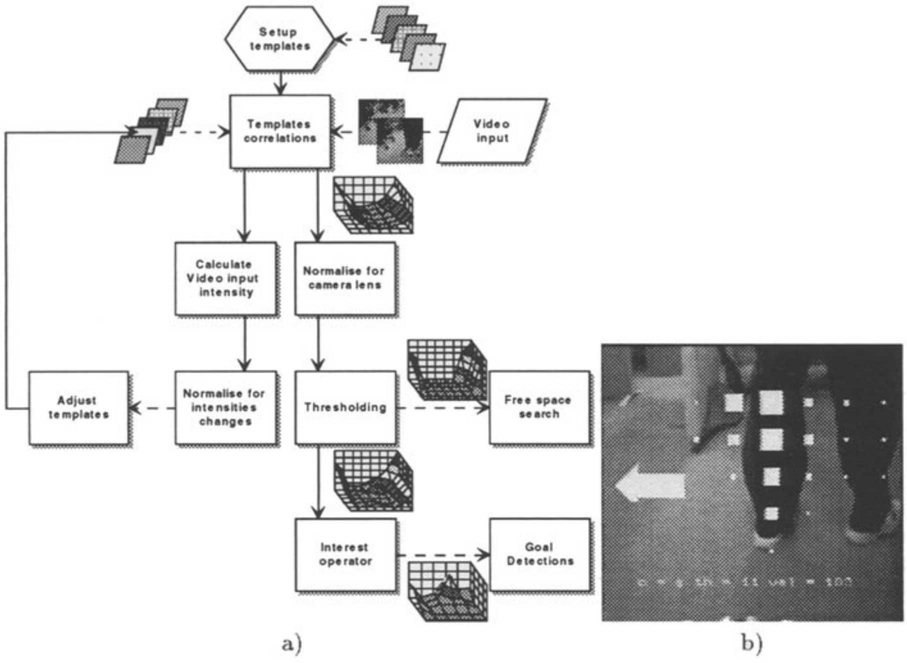


Fig. 2. Vision Processing: a) Processing flow, b) Template Matching

3 Visual Processing

One of the great challenges in robotics is to provide robots with appropriate sensing capabilities, that can supply sufficient information to allow the accomplishment of a task within a reasonable response time. Many sensors such as sonar or infra-red can only provide low-level range information, and cannot provide any kind of goal information. We believe vision can accommodate this deficiency and provide a robot with the wealth of information its needs to accomplish tasks more effectively. By using efficient techniques, real-time performance can be achieved. The vision processing methods we are using are based on cross-correlation template matching. Template matching is a simple and effective method for image processing that allows us to achieve real-time performance. The free space detection process can be summarised as follows. Each image is segmented into a grid of 8 columns and 7 rows, producing 56 individual cells, each cell is 64x64 pixels in size. Each of these cells in the live video is correlated against a stored image of the floor taken at the beginning of each experiment. Due to the lens distortion on the CCD camera that we are using, normalisation is performed on the correlation values. After normalising we perform adaptive thresholding of the correlation values, the result of this process determines the amount of free space available to the robot. The goal-detection process utilises the values produced by the normalisation stage of the free-space detection process. An interest operator is applied to the normalised values, this operator highlights any features for the goal seeking behaviour to focus its attention on that could be interesting. To verify that the correct goal has been found a simple reassurance scheme is used. Each of these operations is explained later in the paper. The flow of processing for lighting adaptation is shown in Figure 2a.

$$D = \sum_{x}^{m} \sum_{y}^{n} \mid g(x-m, y-n) - f(x,y) \mid \tag{1}$$

3.1 Template Matching

For each template match, a correlation value is produced ranging from 0 to 65536. This value determines how well matching occurred (the lower the value the better the match). Figure 2b shows the correlation values using white squares, the better the template match the smaller the squares. These correlation values are calculated by using Equation (1). The basic assumption of this technique is that the floor the robot is to travel is of constant texture. As with template matching a set of stored templates are needed. To reduced storage space, one grid cell of the floor is stored for each row of templates to be matched. This cell is taken from the center of the image of a cleared floor. The correlation between the center and the outer edge of the image can vary due to the distortion of the lens. The normalisation stage of the visual processing combats this problem. Figure 3a shows a clear floor image taken from the CCD camera mounted on top of the robot. Figure 3b shows a plot of correlation values of this image. Figure 3c shows a plot after normalisation.

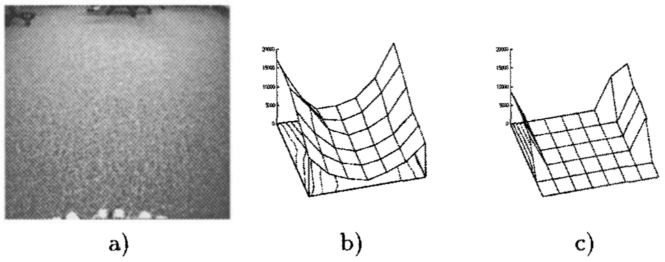


Fig. 3. a) Floor, b) Correlation plot of a floor, c) Normalised correlated values

$$E(x,y) = \begin{bmatrix} e_{(0,0)} & e_{(0,1)} \cdots & e_{(0,n-1)} \\ \vdots & \ddots & \vdots \\ \vdots & \ddots & \vdots \\ e_{(m-1,0)} & \cdots & \cdots e_{(m-1,n-1)} \end{bmatrix} \tag{2}$$

3.2 Normalisation

Empirically it was determined that a polynomial relationship for the correlated values existed, due to the distortion of the camera lens. In Figure 3a, we can see the confidences of each template match plotted by the correlated values for each location of the image. The lower the value, the higher the confidences. To overcome this problem, a polynomial curve was fitted to the plot. For implementation purposes the discrete form shown in Equation (2) was used to perform the normalisation. At each location of the template match, an associated error value is computed.

3.3 Calculating Free-space

Free-space is determine using a threshold value, this is calculated by using the new correlated values produced from the normalisation stage. The threshold value is determined using the variances between all the normalised correlated values, refer to Equation (3). This threshold is then applied to the normalised correlated values, the results are shown in Figure 3b.

$$\sigma = \frac{\sum_n f(x,y) - \bar{f}}{n-1} \tag{3}$$

3.4 Determining velocity

The velocity of the robot is calculated using the amount of free space that is available. The size of free space is determined by the Free-Space behaviour, and is calculated from the ratio of the maximum horizontal and vertical free space available in its view. The calculation is shown in Equations (4) and (5).

$$top_{speed} = max_{speed} \times \frac{y}{max_y} \tag{4}$$

$$velocity = top_{speed} - \left| \frac{top_{speed} \times x}{max_x} \right| \tag{5}$$

3.5 Interest-Operator

The Goal detection scheme in the original system searches the complete images for a given goal, it was noticeably inefficient [Cheng and Zelinsky, 1996]. An interest operator was introduced to increase the efficiency of the goal seeking behaviour by reducing the search space. This interest-operator can also be used as a feature detector that doesn't use an explicit model, and leads to landmark acquisitions. The goal detection processing is as follows. An image from the video camera is shown in Figure 4a. This image is then processed through the correlation step describe previously. This produces the correlation values shown in Figure 4b. Normalisation is then performed on the values, as shown in Figure 4c. The final step involves applying the Interest-operator, its effect is shown in Figure 4d. The Interest-operator produces a single peak that easily identifies a feature of interest in the robot's view, using Equation (6). This calculation has the effect of producing a second derivative of the correlated values.

$$R(t) = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix} \tag{6}$$

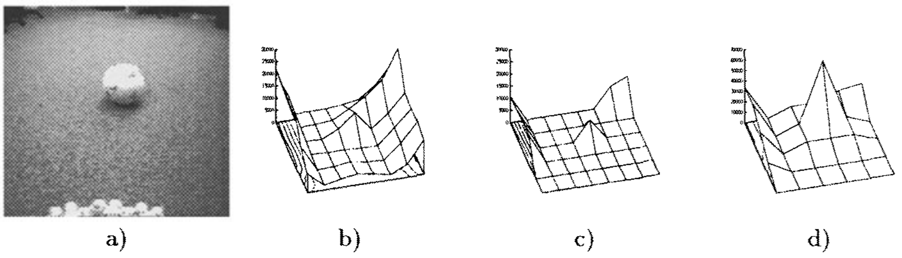


Fig. 4. Interest Operator: a) Soccer ball, b) Correlated Ball, c) Normalised Ball, d) Segmented Ball

3.6 Screen to floor transformation

After determinating where in the video image we would like to move toward, we will need to transpose this location in the image to real world coordinates. Due to the fix configuration of the current camera system, minimum visual coverage is provide to the robot, as shown in Figure 5a. By exploiting this constraint we have derived a simple pixel to floor transformation.

The coordinate transformation is calculated by measuring four distances of the floor, as illustrated in Figure 5b and Figure 5c. The first distance is the height of the camera to the floor, $height_y$. The second distance is the nearest view point of the camera, $blind_y$. The third distance is the furthest visible point to the top view of the camera, $blind_y$ plus $length_y$. The third distance is the furthest point to the left/right of the camera view, $length_x$. With these measurements we can calculate the three angles required, using Equations (7), (8) and (9). From the angles we can determine the angular ratio in both the x and the y directions. Figure 5b shows the side view of the robot. Figure 5c shows the top view of the robot.

$$\alpha = \tan^{-1}\left(\frac{height_y}{blind_y}\right) \tag{7}$$

$$\theta = \tan^{-1}\left(\frac{height_y}{blind_y + length_y}\right) \tag{8}$$

$$\beta = \tan^{-1}\left(\frac{blind_y + length_y}{length_x}\right) \tag{9}$$

The following equation calculates the transformation of the y^{th} pixel to the vertical distance on the floor.

$$y = \frac{height_y}{\tan\left(\theta + \frac{pixel_y(\alpha - \theta)}{screen_y}\right)} + \frac{robot_y}{2} + blind_y \tag{10}$$

where $pixel_y$ is the y^{th} pixel of the screen. $robot_y$ is the length of the robot in the y direction.

The following equation calculates the transformation of the x^{th} pixel to the horizontal distance on the floor.

$$x = \tan\left(\frac{\beta\,(1 - 2pixel_x)}{screen_x}\right) \times y \tag{11}$$

where $pixel_x$ the x^{th} pixel of the screen.

From the Equations (10) and (11). we can calculate the floor space (x, y, θ) for the robot to travel towards from the corresponding point in the image space.

3.7 Lighting Adaptation

A scheme for combating the problem of subtle lighting changes have been incorporated into our system. Others researchers have also experienced problems using vision sensors, Lorigo [Lorigo, 1996] Horswill [Horswill, 1994]. Problems such as the shadow of an obstacle being mis-interpreted as being part of the obstacle. In our implementation, we use the correlation processor to determine the

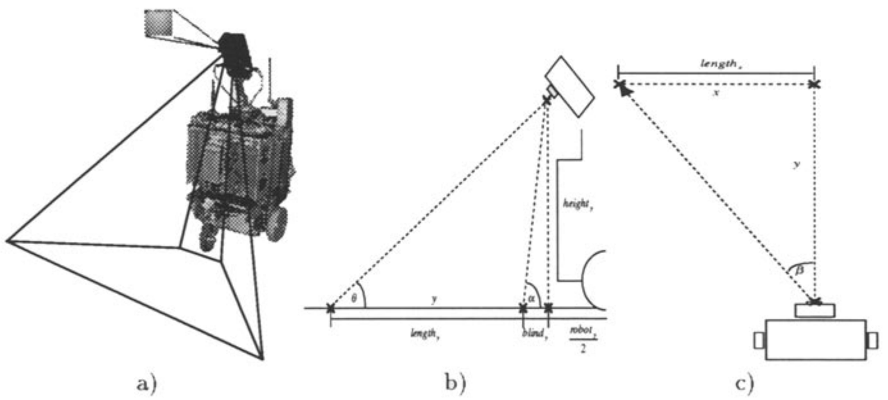


Fig. 5. Camera configuration: a) Coverage, b) Side view, c) Top view

average intensity of an given region in the video steam. This allows the system to dynamically adapt to various lighting condition in real-time.

$$\bar{I}_{now} = \frac{\sum_x^m \sum_y^n \mid f(x,y) - \mathcal{Z}_t(x,y) \mid}{m \times n} \tag{12}$$

where $f(x,y)$ is the greyvalue of the pixel in the last frame
$\mathcal{Z}_t(x,y)$ is a zero template

$$\bar{I}_{new} = \begin{Bmatrix} \mid \bar{I}_{old} - \bar{I}_{now} \mid \leq Sensitivity\ \bar{I}_{old} \\ \mid \bar{I}_{old} - \bar{I}_{now} \mid > Sensitivity\ \bar{I}_{now} \end{Bmatrix} \tag{13}$$

$$\forall (m,n) : g(m,n) = g(x,y) + (\bar{I}_{old} - \bar{I}_{new}) \tag{14}$$

The adaptation process are as follow:

- calculate current image intensity, using Equation (12);
- determine the differences between the old intensity of the initial template in comparison to the new intensity of the current image frame;
- check if these differences are within a sensitivity range, using Equation (13);
- finally adjust each pixel of the template according to this differences, using Equation (14).

Figure 6 shows our system adapting to subtle light changes in the environment. The size of the black squares indicate the level of correlation, after adaptation the correlation improves markedly.

3.8 Recognition/Reassurance

The final stage of the goal detection process, reassures the goal detector that it is looking at the visual cue that it is seeking. Prior to the execution of an experiment a set of templates of the goal (e.g. Ball) are stored. All templates are recorded from various positions from the goal.

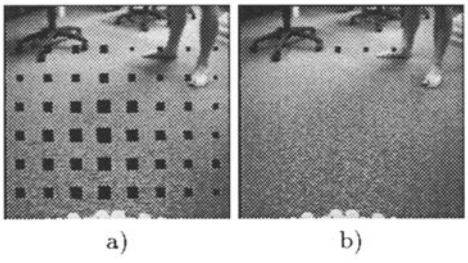


Fig. 6. Adapting to lighting changes: a) before, b) after

3.9 Noise filtering

Filtering is also performed to overcome the problems caused by noisy transmission of the video signals to the vision system. The filter also performs sanity checking on all the commands that are sent to the robot, this checking is done by a voting scheme. This ensures the robot does not get into states such as start-stop motion between consecutive video frames.

3.10 Speed of processing

We deliberately slowed down the processing of video frames, this allowed us to analysis the performance of our overall system. It serves as an indication of the minimum amount processing that is required without a substantial loss in performance and functionality. In normal operation the system runs at 30 Hz (i.e. 30 video frames a second), at 2 Hz the system's performance begins to degrades to the point that it becomes unreliable.

4 Visual Behaviours

We have based our system on three basic visual behaviours, Collision Avoidance, Obstacle Avoidance, and Goal Seeking. The focus of the discussion will be on the vision processing associated with these behaviours. The modular computational structure of a Collision Avoidance behaviour can be simplified into Detection and Avoidance. The detection stage involves determining the availability of free-space in front of the robot for unrestrained motion. If insufficient free space is available the robot suppresses its other behaviours and activates the collision avoidance scheme (e.g. looking for a ball). A situation in which this behaviour can be activated is when the robot is not able to move away from an obstacle, such as a dynamically moving obstacle (such as other robots). Therefore this behaviour acts as a safeguard for the obstacle avoidance behaviour. The Obstacle Avoidance behaviour works by searching for free space within the robot's view of its environment. The robot's basic strategy is to move to where there is free space. The visual processing of the free space search for both of these behaviours will be discussed later. One of the problems with other vision-based avoidance methods (such as optical flow Kröse *et al.* [Kröse et al., 1997]) is their inability to integrate with goal based navigation and the inability to distinguish between a goal and obstacles. The Goal Seeking behaviour exploits the free space searching ability of the vision system. This behaviour allows the robot to focus its attention on searching for a given goal. Once a goal is detected, the robot will perform the associated action for that goal, (e.g. Soccer ball servoing).

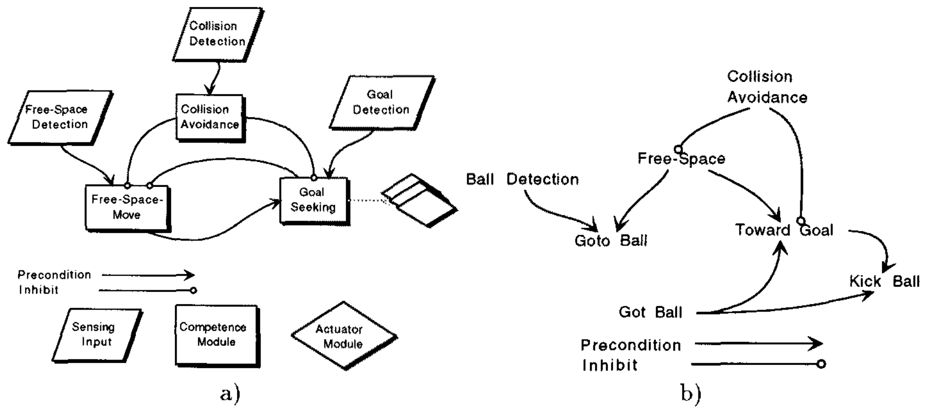


Fig. 7. System Architecture: a) General Architecture, b) Simplified network of soccer behaviours

4.1 System Architecture

The general system control architecture is illustrated in Figure 7. *Detection* modules are depicted as diagonal boxes, actual *behaviour* modules by rectangular boxes and the diamonds are an indication of the *associated actions*. The arrows indicate *preconditions* to a behaviour, and lines with the small circle indicate *inhibitation*. Figure 7 shows that the Collision Avoidance has the highest priority over all the other behaviours. The Free-Space-Move needs to be active when the Goal Seeking behaviour is activated. By adapting the goal seeking behaviour we can perform the appropriate actions.

4.2 Soccer skills

By exploiting the opportunistic property of the Goal Seeking behaviour we have developed basic soccer playing skills for our robot. A simplified network of behaviours is shown in Figure 7b. In this version of the network of behaviours has multiple instances of the Goal Seeking behaviours. i.e. Kick Ball, Toward Goal and Goto Ball exist.

Got Ball Got Ball is determined by visually detecting if a ball is directly in front of the robot.

Goto Ball The Goto Ball behaviour is activated by Goal Detection, once a 'Ball' is detected the robot causing the robot to go towards the ball (visually servoing), as shown Figure 8a-b. The robot avoids obstacles on its way using Free-Space-Move. The motion of the robot is safeguarded by the Collision Avoidance behaviour. This behaviour is an instance of the Goal Seeking behaviour driven by a visual cue, i.e. "The Ball".

Toward Goal This behaviour is triggered by a precondition from Got Ball, and its subsequent motion is determined by the Free-Space-Move behaviour together with its goal to move toward the *Soccer Goal.*

Dribbling The effects of dribbling was achieved through the combination of Got Ball detection, and Free-Space-Move.

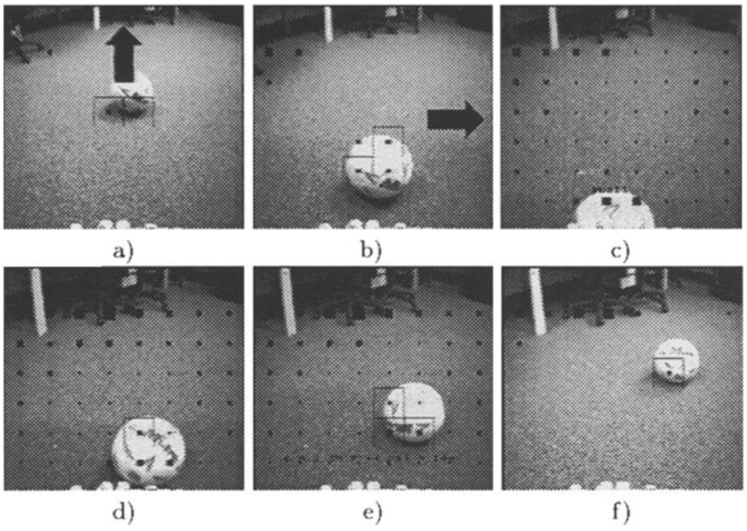


Fig. 8. Kicking

Kick Ball This behaviour simply waits until the *Soccer Goal* are sufficiently close enough, then it activates a kick action. That is done by stopping then accelerating rapidly, producing a kick like action. Figure 8 shows visual servoing and then the kicking action by our robot.

5 Conclusion and Further Work

The system we have presented here demonstrated a set of basic soccer playing behaviours for a mobile robot. The was achieved through the use of local vision.

All of the behaviours exhibit real-time performance that are robust and reliable in both static and dynamic environments. In normal operation the current system can perform its task at a robot velocity of up to 0.5 m/sec. Our aim is to move the robot at 1 m/sec.

Image correlation is an efficient method for achieving real-time vision processing. Although our system performs visual processing using special hardware, we believed that the vision processing we have described is well within the reach of current commercial processors. This has inspired us to begin the construction of on-board vision processing modules for our future works.

Presently our work has been primary focused on the development of robust vision based soccer skills for a mobile robot. The system in its current states does not conform with the rules that has been established in the present RoboCup. Further work will be gear toward a colour based image correlation system. This will allow our system to conform with the RoboCup specifications. Colour template matching will also enhance the reliability of our system.

Acknowledgements

This research is funded by a grant from the Australian Research Council and supported by Wind River Systems supplier of the VxWorks® operating system.

References

[Brooks, 1986] Brooks, R. (1986). A Layered Intelligent Control System for a Mobile Robot. In *IEEE Trans. on Robotics and Automation, RA-2.*

[Cheng and Zelinsky, 1996] Cheng, G. and Zelinsky, A. (1996). Real-Time Visual Behaviours for Navigating a Mobile Robot. In *Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 973–980. IROS'96.

[Horswill, 1994] Horswill, I. (1994). Visual Collision Avoidance by Segmentation. In *Intelligent Robots and Systems*, pages 902–909. IROS'94.

[Iida and Yuta, 1991] Iida, S. and Yuta, S. (1991). Vehicle Command System and Trajectory Control for Autonomous Mobile Robots. In *Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems*, Osaka, Japan. IROS'91.

[Kitano et al., 1997] Kitano, H., Asada, M., Kuniyoshi, Y., Noda, I., and Osawa, E. (1997). RoboCup: The Robot World Cup Initiative. In *Proceedings of the first International Conference on Autonomous Agents*, pages 340–347, Marina del Rey, CA.

[Kröse et al., 1997] Kröse, B., Dev, A., Benavent, X., and Groen, F. (1997). Vehicle navigation on optic flow. In *Proceedings of 1997 Real World Computing Symposium*, pages 89–95.

[Lorigo, 1996] Lorigo, L. M. (1996). Visually-guided obstacle avoidance in unstructured environments. Master's thesis, Massachustts Institute of Technology.

[Yuta et al., 1991] Yuta, S., Suzuki, S., and IIda, S. (1991). Implementation of a Small Size Experimental Self-Contained Autonomous Robot. In *Proceedings of the 2nd Int. Symposium on Experimental Robotics*, pages 902–909, Toulouse, France.