A Global Vision System for a Robot Soccer Team

David Ball, Gordon Wyeth, Stephen Nuske
School of Information Technology and Electrical Engineering
The University of Queensland, Australia
{dball, wyeth}@itee.uq.edu.au

Abstract

This paper describes the real time global vision system for the robot soccer team the RoboRoos. It has a highly optimised pipeline that includes thresholding, segmenting, colour normalising, object recognition and perspective and lens correction. It has a fast 'paint' colour calibration system that can calibrate in any face of the YUV or HSI cube. It also autonomously selects both an appropriate camera gain and colour gains robot regions across the field to achieve colour uniformity. Camera geometry calibration is performed automatically from selection of keypoints on the field. The system acheives a position accuracy of better than 15mm over a 4m × 5.5m field, and orientation accuracy to within 1°. It processes 614 × 480 pixels at 60Hz on a 2.0GHz Pentium 4 microprocessor.

1 Introduction

This paper describes the real time global vision system for the Small Size League (SSL) robot soccer team, the RoboRoos. One research emphasis in this league is promoting fast, accurate overhead vision systems that are robust to unknown and varying lighting conditions. As the global overhead vision system is the primary sensor for the robots, it is critical that it be robust for success in the competition. It provides the robots with their location and orientation and the location of the ball and the opponent robots.

The difficulties for a vision system within this league are:

- coping with unknown and varying lighting conditions at the venue,
- the small amount of setup and calibration time,
- non-uniformity of robot colours and markings in different teams,
- non-uniform brightness across the field and potential for sharp shadows,
- identifying a ball that moves at over 5 m/s, and robots that reach speeds of 3 m/s.

This vision system demonstrates the ability to cope with these difficulties using two FireWire cameras and a single laptop. It has a novel approach to improve the speed and accuracy at which colours can be classified while not requiring real time colour conversions. It also has a novel approach to handle both global brightness changes during game time and non-uniform colour intensity across the field including sharp shadows.

This paper is structured as follows. The first section introduces the paper, the vision system and the testing domain. The second section points towards other research within the SSL domain. The next section describes the system in detail and presents relevant results. Section four discusses the results and presents some possible future directions for the system. Lastly section five concludes the paper.

1.1 RoboCup

The RoboRoos vision system is applied to the Small Size League of the RoboCup competitions that are held annually. In this league, both teams have five robots that each must physically fit inside a cylinder with a diameter of 180mm and a height of 150mm. Devices to dribble and kick the ball are permitted as long as they do not hold the ball and 80% of the ball is kept outside of the convex hull of the robot. The dimensions of the field are 4 x 5.5 meters, with an orange golf ball acting as the soccer ball. The rules are similar to the human (FIFA) version of the game, with exceptions such as the elimination of the offside rule and changes required to make sense for wheeled robots. The robots are fully autonomous in the sense that no strategy or control input is allowed by the human operators during play.

1.2 RoboRoos

The University of Queensland's robot soccer team, the RoboRoos [Ball *et al.*, 2003], [Wyeth *et al.*, 2002], [Wyeth *et al.*, 2001] and [Wyeth *et al.*, 1999] is one of the longest standing teams in the small-size league of RoboCup having competed annually since 1998. During these years many research areas have been explored especially in the areas of multi-robot coordination and navigation in dynamic environments. The RoboRoos came second at RoboCup 2003 and 2004 beaten both times by only a single goal in the final.

The RoboRoos system is a layered set of subsystems, where each subsystem performs a different task. There are two Basler FireWire cameras, one mounted over each half of the field capture global images of the field. The vision system processes the images to identify and locate the robots and the ball. This state of the field is sent to the Multi-Agent Planning System [Tews, 2002]. MAPS coordinates the RoboRoos by selecting a behaviour for each robot. The MAPS behaviours are interpreted by the AES (Action Execution System) system. Each behaviour has a set of appropriate parameters and a notion of the overall desired robot motion. The Navigation [Browning, 2000] module attempts to achieve the desired motion behaviour while avoiding obstacles. The Navigation module determines the immediate desired heading and distance for the Motion System. The Motion system accelerates and decelerates the robot to the desired heading and distance by creating force limited trajectories.

2 Literature

Carnegie Mellon University's F180 vision system [Bruce and Veloso, 2003] demonstrates an approach using different coloured markers for identification and orientation. They have demonstrated a statistical approach to determining the size and number of markers required in an environment where determining the 'truth' to determine error from can be difficult.

Free University of Berlin, Germany [Egorova *et al.*, 2004] demonstrates a vision system capable of autonomous colour and geometric calibration. The system automatically defines the colour maps that store the parameters for each important colour in a grid that is superimposed on the field. Geometric calibration uses gradient descent to determine the placement of field vertices to sub-pixel accuracy.

3 The RoboRoos Vision System

3.1 Overview

The section details the RoboRoos vision system. It gives and overview of the flow of information in the system and goes on to detail each layer in the pipeline. For each system the average time and percentage of the total time is given. To give an idea of the difficulty for the vision system a screen shot of the field as seen by the cameras is shown in Figure 1. This was taken from the RoboCup 2004 competition that was held in Lisbon, Portugal.

The RoboRoos vision system pipeline is shown in Figure 2. The visual sensor is two Basler A301fc FireWire cameras, one mounted over each half of the field. These cameras are each capable of 640x480@80FPS but due to the FireWire bus bandwidth limitation they are used at approximately 614×470 pixels at 61.7FPS giving a frame time of 16.4 milliseconds. The processing time for each module is given as it is described. Any remaining time is taken by the overhead of receiving the images from the cameras and displaying to the screen. A pixel represents approximately 6.6 millimetres on the field. Each image is processed separately until after the lens and perspective correction stage.

The first step is to convert the Bayer RGGB data into an RGB image. The pixels are colour classified by a lookup table to find seeds for growth into regions of interest. Non-background pixels are thinned, and then the remaining pixels are grouped into potential ball and robot regions. The object identification layer parses these regions to find the ball and the robots. The image coordinates are then perspective and lens corrected. Lastly the information from both cameras is combined.

The locations and orientation of our robots and the locations of the opponent robots and the ball is sent via a network socket to the Intelligence system. This system begins by Kalman filtering the incoming data from the vision system. This minimises the effect of losing robots and pixilation issues that affect velocity calculation.

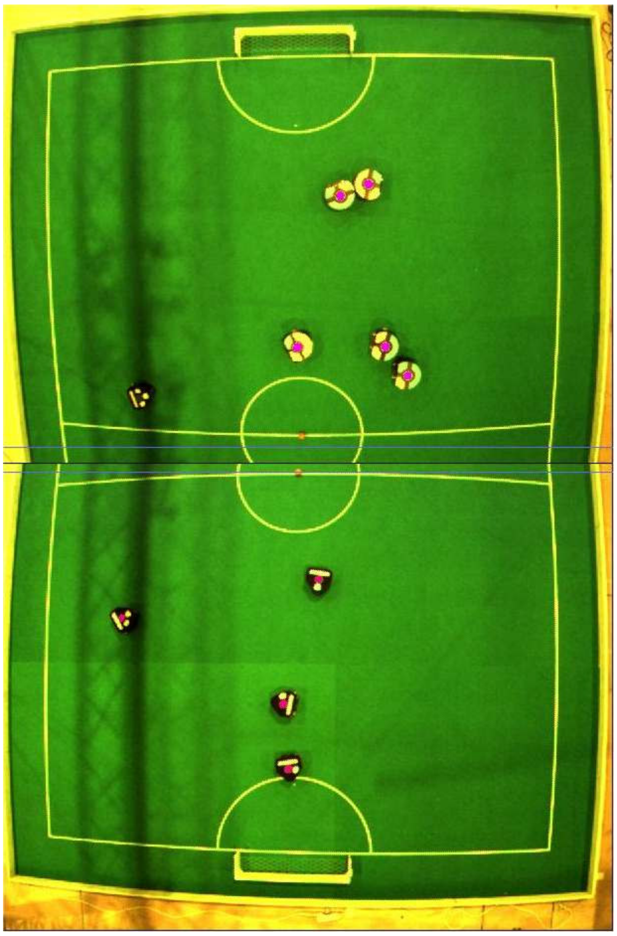

Figure 1. Screenshot showing the cameras point of view. The images are shown joined together although they are generally treated as separate images. This picture is taken from Lisboa, Portual at RoboCup2004. In particular note the sharp shadow across the field. This shadow is caused by the camera mounting bar. In this image the blue and yellow markers that vision has found have been painted over in pink. This is a full resolution image.

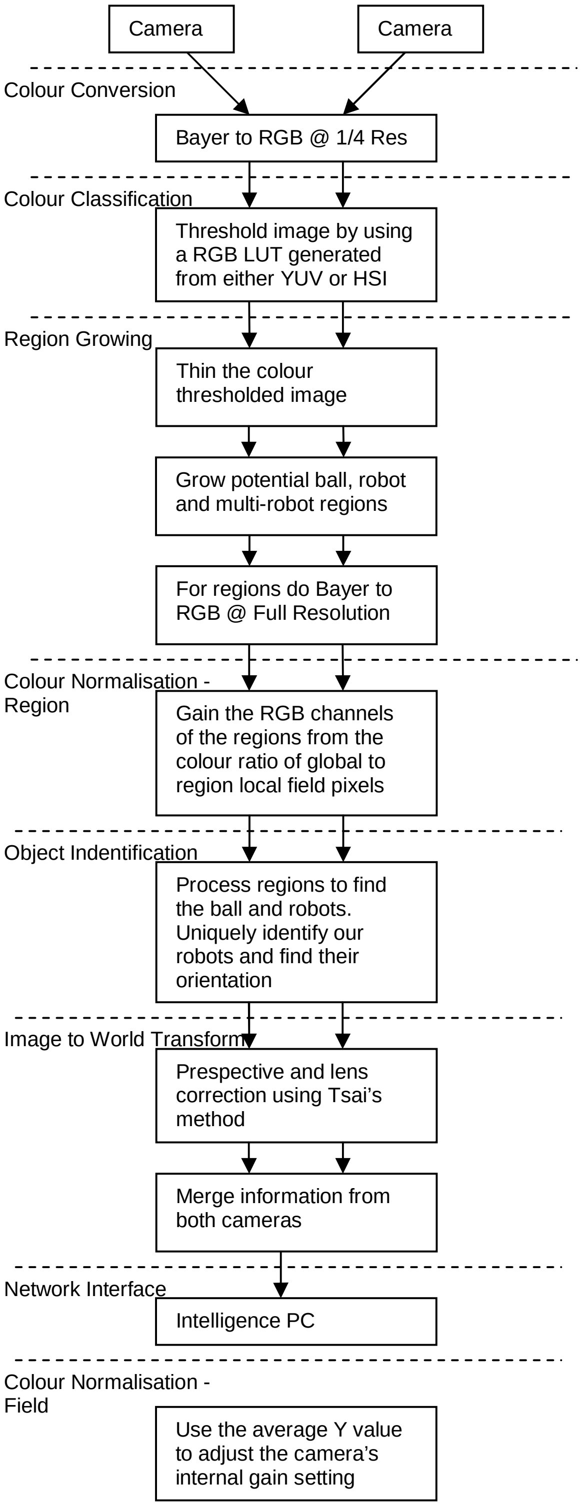

Figure 2. Pipeline diagram for the RoboRoos 2004 vision system. Images are treated seperately until the merging module.

3.2 Colour Conversion

The colour conversion layer converts the cameras proprietary Bayer 2G format into the RGB format. Figure 1 shows an image from RoboCup 2004. The image is originally converted at only quarter resolution to improve real time performance. The RGB image format is 32 bits where there are 8 bits for each of the three colours (0 - 255) and last byte is used to indicate whether or not the pixel has been classified and is valid. This layer takes 0.78 milliseconds or 4.77% of the total time to process the frame.

3.3 Colour Classifier

In this layer the quarter resolution RGB image is organised into several base colours using a look up table. Figure 3 shows the field thresholded at full resolution. The image is classified into the following groups:

- Field
- Ball
- Blue Marker
- Yellow Marker
- Black
- White
- Unknown (Show as pink)

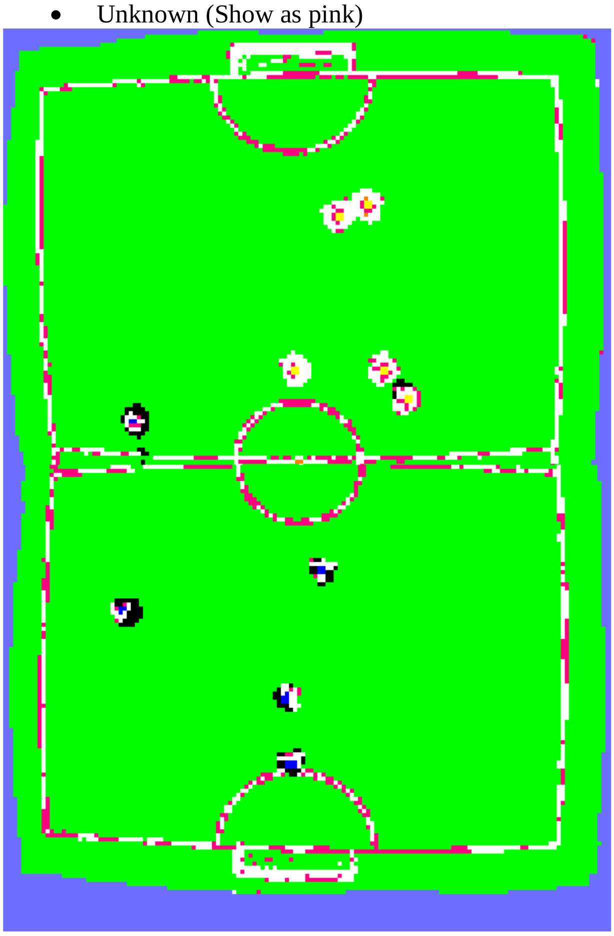

Figure 3. Colour classified image at quarter resolution. This image is then region connected to find potential robot and ball regions. Note that regions outside of the field have been masked out from this process (in light blue).

The colour groups can be specified in either YUV or HSI colour spaces, even though the image is in the RGB format. Rather than converting the image to the colour space used for calibration, the LUT is generated by converting the YUV or HSI colour groups for all possible RGB pixels. Not having to convert the image from RGB to an alternative colour space dramatically improves the performance of the system. This layer takes 3.92 ms or 23.88% of the total time.

The process for determining the colour groupings is to first select examples from around the field of the colours defined above. The example pixels are then shown in their UV positions (assuming YUV is used as the classification basis). These can then be grown automatically into colour regions using a dilation algorithm. Generally though the regions are hand tuned to achieve optimal thresholding. Figure 4 shows the colour classification calibration window in UV, an image and the resultant colour classified image in quarter resolution. The colour regions in the UV plane are initially projected vertically along the Y axis. The user can edit the colour volumes further in the YU and YV planes to adjust colour classification for different brightness levels.

While the colours within the field are defined the colour outside are not. To stop unnecessary colour classification and potential false region growing the outside of the field is masked out. This prevents both the colour classification and region growing layers from using these pixels.

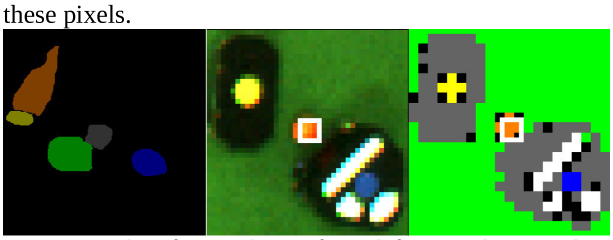

Figure 4. This figure shows from left to right a) colour groupings in UV, b) raw image, c) the final colour classified image in quarter resolution. The black pixels are represented in grey and the unknown pixels in black.

3.4 Region Growing

This layer grows regions from the colour classified image to be processed by the object identification layer. Before the regions are grown the images are morphologically thinned, using the field (green) pixels as background, and all other pixels as pixels of interest. This removes any noise and helps to remove field lines.

The next module now grows regions by connecting pixels together. These regions are queued for the object identification layer to process. Three different regions are grown:

- Potential ball regions. Orange pixels.
- Potential robot regions. Blue, Yellow, Black, White pixels.
- Potential multi-robot regions. Robot regions that could potentially contain multiple robots.

Typically due to the Bayer conversion there are many orange pixels between the white field lines and the green field pixels. While the object identification will process all potential regions and return the best ball it significantly improves the performance of the system to minimise the number of regions for the next layer to process. Consequently, the orange pixels that lie along field lines are ignored in the region growing process. Figure 5 shows the output of this module. The region growing module takes 2.62 milliseconds or 15.95% of the total time. Note that the black pixels show pixels that have been thinned, while the red pixels show regions of interest for further processing.

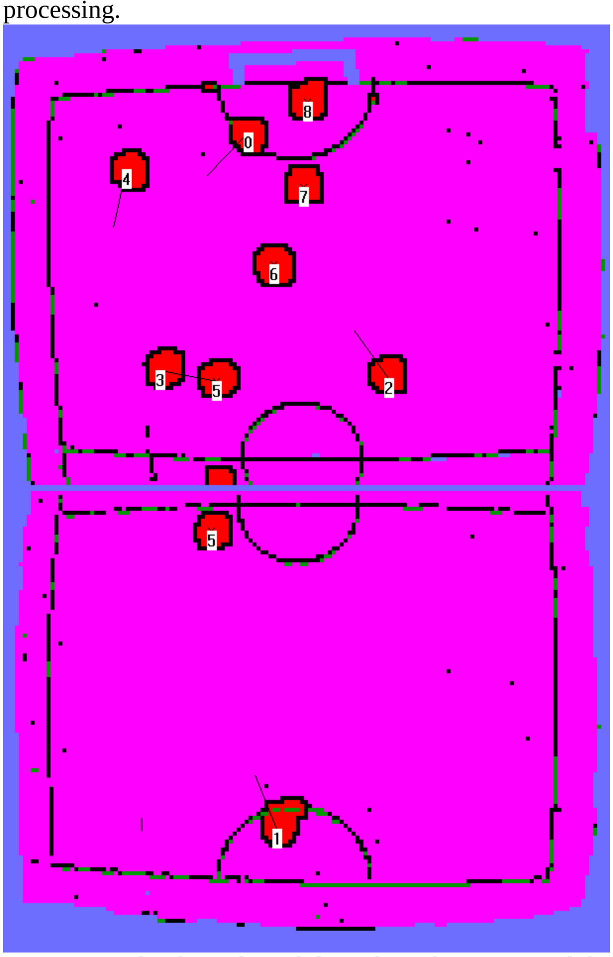

Figure 5. Isolated pixels and lines have been removed by thinning (black pixels). Regions of interest have been grown by connecting pixels (red pixels). Note that there is no ball in this image, but the module has found two potential balls at the top of the image near the goal.

3.5 Colour Normalisation - Region

Some non-uniformity will always occur due to the wide angle lens taking in less light towards the edge of the image, but more serious problems arise from the shadowing of light sources. This layer is responsible for handling non-uniform lighting across the field. Each region of interest is normalised by comparing the colour values in the region to the average colour values of the entire field. Both of these effects can be seen in Figure 1.

Two methods of normalisation were tested. The first involves using only the Y values to determine the normalisation ratio. This ratio is then applied to each RGB channel individually. The second method is to determine individual ratios for each of the RGB channels. To test this layer a robot was programmed to follow the path through a sharp shadow as shown in Figure 6. Table 1 shows the identification percentage results over the time of the test.

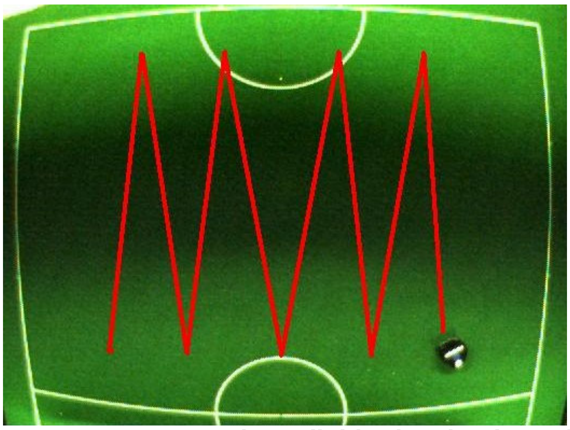

Figure 6. Approximate path travelled by the robot during the colour normalising - region testing. This path is designed to drive the robot through the darker region at multiple locations.

Table 1. The table shows the percentage of robot identifications for the robot travelling in the path shown in Figure 6.

Colour Normalising State	Identification Percentage
None	62.3%
Y Normalising	80.9%
RGB Normalising	95.4%

Table 1 demonstrates the performance increase of the system in handling non-uniform lighting conditions that can be obtained by normalising regions of the image. In particular it demonstrates that normalisation on a channel by channel basis is the most effective for dealing with shadowed regions. This layer takes 3.21 milliseconds or 19.57% of the total time. (There is also 2.13 milliseconds or 13.01% of the time for the local full resolution Bayer to RGB conversion.)

3.6 Object Identification

In the object recognition layer the grown regions are processed to find the robots and the ball. Potential ball regions (regions containing orange pixels) are processed first to find the best ball. The best ball is the object that best meets criteria of an appropriate number of orange pixels and the roundest shape (see below). Regions big enough to be multiple robots are processed next and the rest processed for single robots.

Identifying the robots requires finding the location and colour of the central identifying marker; one team has blue markers, and the other yellow. For robots on our team we analyse the markers further to find the identity of the different team members, and the orientation of the robot. The identity of our robots is represented in binary using approximately square markers. (Zero is represented by binary 7). Robot 3 is shown in Figure 4.

Second moment of area analysis is a central theme in the object identification layer. The second moments are used to find the orientation of the long white stripe that runs along the front of the robot. The first problem though is determining which markers are identification markers, and which is the stripe that gives orientation. To solve this we use a measure based on the second moments that we have called "stripiness". The second moments are defined as (where the c denotes the centre of the object):

$$J_{xx} = \sum (x - x_c)^2 = \sum x^2 - x_c \sum x$$
$$J_{yy} = \sum (y - y_c)^2 = \sum y^2 - y_c \sum y$$
$$J_{xy} = \sum (x - x_c)(y - y_c) = \sum xy - x_c \sum y$$

The stripiness, s, of an object can then be determined by:

$$s = \frac{J_{xx} + J_{yy} - \sqrt{(-J_{xx} - J_{yy})^2 - 4(J_{xx}J_{yy} - J_{xy}^2)}}{J_{xx} + J_{yy} + \sqrt{(-J_{xx} - J_{yy})^2 - 4(J_{xx}J_{yy} - J_{xy}^2)}}$$

This measure becomes large for single strips of pixels, and approaches one for round groups of pixels. Stripiness is used not only to find to long thin marker stripe at the front of the robot, but is used throughout marker and ball identification to ensure that the groups of pixels under analysis are sufficiently round to be a marker or a ball.

Once the stripe at the front of the robot has been found using the stripiness measure its orientation is also determined using the second moment of area measures. The orientation, θ, is found by:

$$\theta = \operatorname{atan2}(2J_{xy}, J_{xx} - J_{yy})$$

Another significant part of this module is accounting the difference in size between markers in the centre of the images and on the outside. Table 2 shows the number of pixels in the orientation stripe, identity markers and centre markers between the centre, side and corner of the field. This table shows significant difference between markers at different locations. It shows that determining thresholds to distinguish between a stripe on the corner and an ID marker in the centre can be problematic. To account for this the number of pixels in the markers in the centre and corner of the image was recorded. Then each robot region has its marker size thresholds determined using a linear mapping between the centre and the corner. Note that both minimum and maximum thresholds are determined.

Table 2 Number of pixels in markers at different locations in the image. Note the large difference in size between the centre and the outside.

Robot Location	Stripe Pixels	White ID Marker Pixels	Centre Marker Pixels
Centre	84	33	53
Side	32	19	22
Corner	22	10	12

The object identification module takes 2.94 milliseconds or 17.91% of the frame time.

3.7 Image to World Transformation

This layer corrects the locations and orientations from image to world coordinates and merges the information from the two images. The image to real world correction uses an implementation of Tsai's [Tsai, 1987] camera calibration technique. To set the conversion parameters the user clicks on points across the field. 18 calibration points spread evenly across the field were used in 2004. Tsai's method corrects individual points for camera alignment and perspective effects, but does not provide any direct ways of correcting a robot's orientation. This is achieved by determining the image position at a set distance from the robot as its orientation. This is corrected into world coordinates using Tsai's algorithm. The angle between the robot's world location and this location is the world orientation. This module takes 0.27 milliseconds or 1.63% of the total frame time.

To determine the accuracy of the Perspective and Lens correction module the robot's real position is measured and compared to the position given by the vision system. Table 3 shows the results of location testing for five locations spread over the field. Table 4 shows the results of orientation testing at the same five locations at 3 different orientations.

Table 3. Results for the Perpective and Lens Correction Module in regard to location. Note that a pixel represents approximately 6.6 millimetres on the field. This table shows that this module is able to correct to real world locations with a high degree of accuracy.

	Real World (mm)	Corrected (mm)	Uncorrected (mm)
Pos 1 x	400	402	291
Pos 1 y	3400	3401	3511
Pos 2 x	0	-7	-97
Pos 2 y	3000	2990	3095
Pos 3 x	400	393	243
Pos 3 y	3000	3008	3165
Pos 4 x	1300	1293	1254
Pos 4 y	3400	3406	3601
Pos 5 x	1300	1297	1241
Pos 5 y	2800	2813	3116

Table 4. Results for the Perspective and Lens Correction module in regard to orientation. This table shows that this module is able to correct to real world orientations with a high degree of accuracy.

	Real World (deg)	Corrected (deg)	Uncorrected (deg)
Pos 1	0	0.05	-8.7
Pos 1	30	31.5	33.45
Pos 1	45	44.48	53.44
Pos 2	0	1.4	-12.8
Pos 2	30	32.1	21.38
Pos 2	45	45.44	42.45
Pos 3	0	359.37	-5.7
Pos 3	30	31.58	28.85
Pos 3	45	44.22	51.69
Pos 4	0	0.1	-4.81
Pos 4	30	30.9	32.23
Pos 4	45	45.98	52.5
Pos 5	0	0.915	-1.12
Pos 5	30	30.77	31.7
Pos 5	45	45.58	46.32

After the conversion to world coordinates, the locations and orientations of the robots from the separate cameras must be merged. Note that opponent robots are not uniquely identified by the object recognition layer. This is problematic if opponent robots are near the centre of the field where they can be seen in both camera images. If the merging module finds two opponent robots that are within a predefined threshold distance they are treated as the same robot and their position is averaged. This module takes on average 0.27ms or 1.63% of the total frame time.

3.8 Colour Normalising – Field

This layer is responsible for handling dynamic changes in light during a game. It does this by attempting to maintain a constant average field colour intensity by adjusting the camera's internal gain. This will affect the gain of the next image that is sent by the camera. The first step is to determine the average Y value of the field by averaging every 16th pixel. If this average Y is not within a predefined threshold range the cameras internal gain is either incremented or decremented accordingly. This is done after the state of the field is sent to the Intelligence PC to minimise overall latency. The time for this layer to complete is very short.

To show the effect of this module the percentage of robots identified was recorded at 5 minute intervals during light and dark times during the day. The system was calibrated only once at 8:00pm and left running for twenty-four hours. Figure 7 shows the performance without the field colour normalising and Figure 8 shows the performance with the colour normalisation module. The module is able to maintain a constant Y value which directly affects the percentage of robots detected. Note the inverted characteristics of the Camera Gain and Average Y between the two tests.

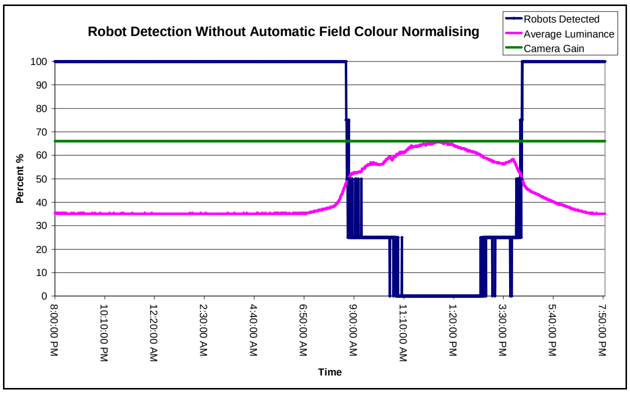

Figure 7. System performance at day and night without colour normalising - field. The system was calibrated only once at the start of the period.

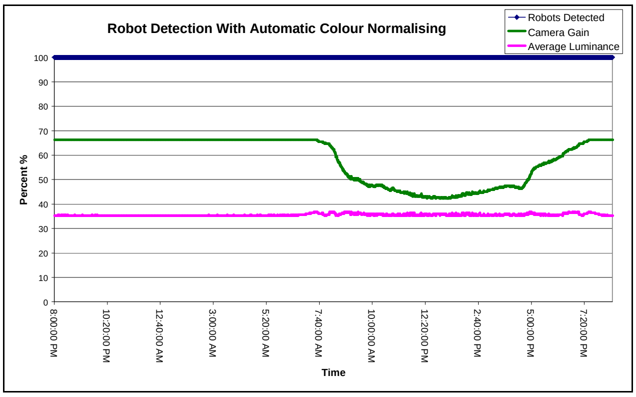

Figure 8. System performance during day and night with colour normalising - field. The module demonstrates the ability to maintain a constant average Y value by adjusting the cameras internal gain.

4 Discussion

This paper has stepped through the layers of the RoboRoos 2004 vision system providing relevant results. The layered approach assists in determining why an object is not being identified or why it is being incorrectly identified. It also gives robustness through competency at each layer.

This year saw the integration of modules to handle both non-uniform and dynamic lighting. The results show that these modules work and are able to handle a wide range of static and dynamic lighting conditions autonomously.

Stripiness has proven a useful measure to determine that an object is of appropriate shape. It is especially important in disguising between identification markers and the orientation stripe.

The Image to World Transform layer is able to locate and in the case of our robots determine their orientation to a high degree of accuracy. Location accuracy is important not only for defensive formations but also increases the ability of the robot to acquire the ball. Orientation accuracy is vitally important for precision shooting and passing.

This system is tested at the annual RoboCup competitions. It forms the primary sensor for the robots and the only sensor to give them global localisation therefore it is vitally important that it is a robust system. The system was shown to work well at RoboCup2004 being one of only a few that could robustly handle the dynamic lighting and non-uniform colour intensity across the field. It was also shown to be one of the fastest vision systems to setup and calibrate by one of the smallest teams (human members) at the competition. The RoboRoos 2004 vision system works well in the visually harsh environment for which it is designed.

4.1 Future

Future work on the RoboRoos vision system will focus in two areas. The first is combining the velocity filter with the vision system and some common sense. This will help to identify regions that vision cannot determine the identity of and reject false classifications. This is particularly important for the ball which can be completely occluded by a robot.

The second is to add more automated calibration support. Potential areas for automated calibration include:

- Autonomously find the image pixels for perspective and lens correction.
- Autonomously select global individual RGB gains to give optimal separation of the colour groups.
- Autonomously determine the colour regions for the colour classifier for the thresholding module.

5 Conclusion

This paper has described a global vision system capable of quickly and easily determining colour classification groups, handle dynamic light changes, non-uniform colour intensities across the field and accurately identify and locate objects on the field. It runs in real time on a P4 2.0GHz PC and has been tested at the RoboCup 2004 competition with successful results. Robustness in the system is achieved by the layering of competent modules that each handle a specific problem. This system can be completely setup and calibrated for competition in a short amount of time by one operator.

Future work will focus on more automation in the calibration process and the combination of the velocity filter with the reactive vision system.

Acknowledgements

Acknowledgements go to the following students who have worked on the RoboRoos vision system since the beginning: Leighton Carr, Chris Nolan, Adrian Ratnapala, Brett Browning and Ben Brown.

References

[Ball et al., 2003] David Ball, Gordon Wyeth, et al. Design of a Team of Soccer Playing Robots. Autonomous Minirobots in Research and Edutainment (AMiRE), Brisbane, Australia, 2003.

[Browning, 2000] Brett Browning. Biologically Plausible Spatial Navigation for a Mobile Robot. PhD dissertation. Department of Computer Science and Electrical Engineering, University of Queensland, 2000.

[Bruce and Veloso, 2003] James Bruce and Manuela Veloso. Fast and Accurate Vision-Based Pattern Detection and Identification. ICRA'03, the 2003 IEEE International Conference on Robotics and Automation, Taiwan, 2003.

[Egorova et al., 2004] Anna Egorova, Mark Simon, et al. Plug & Play: Fast Automatic Geometry and Color Calibration for Camera Tracking Robots. RoboCup2004 Symposium, Lisboa, Portugal, 2004.

[Tews, 2002] Ashley Desmond Tews. Achieving Multi-Robot Cooperation in Highly Dynamic Environments. PhD dissertation. School of Computer Science and Electrical Engineering. Brisbane, University of Queensland, 2002.

[Tsai, 1987] R. Y. Tsai. "A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses." IEEE Journal of Robotics and Automation, vol. 3: 323-344, 1987.

[Wyeth et al., 2002] Gordon Wyeth, David Ball, et al. UQ RoboRoos: Achieving Power and Agility in a Small Size Robot. RoboCup 2001, 2002, Springer-Verlag.

[Wyeth et al., 1999] Gordon Wyeth, Brett Browning, et al. UQ RoboRoos: Preliminary Design of a Robot Soccer Team. Lecture Notes in AI: RoboCup '98, 1604, 1999.

[Wyeth et al., 2001] Gordon Wyeth, Ashley Desmond Tews, et al. UQ RoboRoos: Kicking on to 2000. RoboCup-2000: Robot Soccer World Cup IV, 2001, Springer-Verlag.