Fast and Robust Stereo Object Recognition for Spheres

Robby McKilliam and Gordon Wyeth
Information Technology and Electrical Engineering
University of Queensland, Australia
wyeth@itee.uq.edu.au

Abstract

The paper presents a fast and robust stereo object recognition method. The method is currently unable to identify the rotation of objects. This makes it very good at locating spheres which are rotationally independent. Approximate methods for located non-spherical objects have been developed. Fundamental to the method is that the correspondence problem is solved using information about the dimensions of the object being located. This is in contrast to previous stereo object recognition systems where the scene is first reconstructed by point matching techniques. The method is suitable for real-time application on low-power devices.

1 Introduction

Recently, stereo imaging has gained large interest for the purpose of scene reconstruction [Leung *et al.*, 2004], and automatic camera parameter estimation [Hartley and Zisserman, 2004; Faugeras, 1992]. Stereo imaging also has a powerful ability for object recognition. The usual approach to stereo object recognition is to use it solely for the acquisition of object position. In this approach mono-vision recognition techniques are used to locate the object in both cameras. The object can be matched in both cameras and its world position found by triangulation. A problem with this approach is that mono-vision recognition techniques are used to find the objects. This paper will show that it is possible to exploit stereo vision to greatly improve object recognition performance with minimal computational overhead.

Another approach to stereo object recognition is to first reconstruct the 3D scene using stereo vision techniques, then find where the object best fits into the reconstruction. This has been the typical approach for existing stereo object recognition systems [Sumi *et al.*, 2002; Rygol *et al.*, 1991]. These systems can locate objects and identify their orientation in cluttered scenes, even when the object in partially occluded, but suffer greatly from the initial need to reconstruct the scene. This process is both computationally expensive and sensitive to noise. The fundamental difficulty with scene reconstruction is that of correctly matching points between the two images. This is known as the *correspondence problem*.

It would seem that attempting to reconstruct the entire scene in order to locate a particular object is unnecessary, for example [Sumi *et al.*, 2002]. Dijck and Heijden [2002] used edge detection to find feature points in each image. The feature points were matched together in a many-to-many fashion. A number of these matches would be incorrect. *Geometric hashing* [Wolfson and Rigoutsos, 1997] was used to efficiently locate the object within the matches. Geometric hashing also rejected incorrect matches.

An interesting property of Dijck and Heijden's technique is that the scene itself is never explicitly reconstructed. By allowing incorrect matches, Dijck and Heijden, allowed the possibility of multiple different scenes. The correspondence problem could be solved during the object recognition stage. The fact that the size and shape of the object being found were known, aided in the solution of the correspondence problem whilst simultaneously locating the object.

This property of stereo object recognition is the main focus of this paper. It is the key to efficient and robust object recognition using stereo-vision. It separates the methods proposed in this paper from the previous approaches to stereo using mono-vision recognition and the stereo-reconstruction techniques. Where Dijck and Heijden noted the property briefly, we make it a fundamental property.

The method proposed in this paper has the following key properties:

- Suitable for real-time application on embedded devices
- Only low resolution is needed to achieve excellent results at reasonable range.

- The recognition is significantly robust against image noise and poor camera calibration. Moreover, image noise minimally increases the computation time.
- The described implementation (section 3) uses region growing based segmentation. This enables the use of colour in a simple and efficient way.

An example of this method was implemented on a humanoid robot to find a ball, poles and a goal (section 3). The results in this paper are taken from tests using the humanoid robot (section 4). Focus is placed on results for the ball (sphere) as this does not require assumptions about object rotation that were relevant to the humanoid only.

2 Methodology

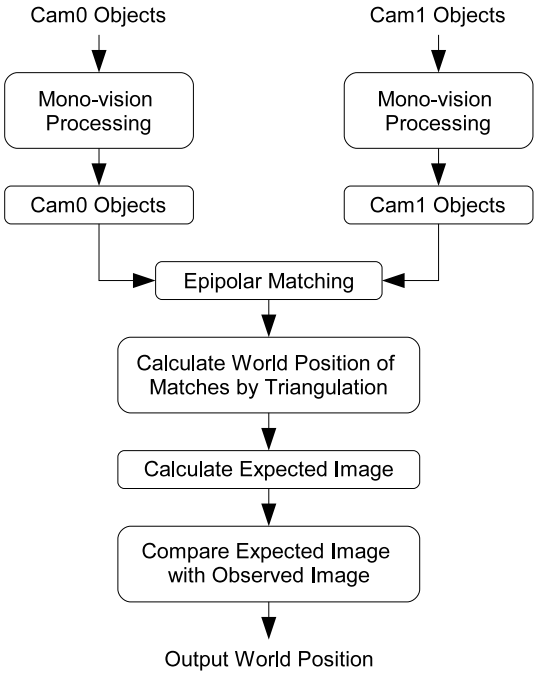

Figure 1: Methodology flow diagram

Figure 1 is a flow diagram of the methodology. This is summarised in the following paragraph:

Given that a number of potential objects have been located in both images, the algorithm matches all the potential objects that satisfy the epipolar constraint. For each match, a corresponding world position can be calculated using triangulation. Given that the dimensions of the object are known, the expected image of the object can be projected onto the cameras using the world position. This is, given that the position and size of the object is known, it's shape can be calculated for each camera. This image can be compared with the actual images of the potential objects in both cameras. The match of potential objects that gives the best comparison is said to be the object.

This methodology is fundamental to the stereo recognition system, we now rephrase it and introduce some notation. Mono-vision image processing returns potential objects from both images. Any type of image segmentation technique could be used. The objects are matched across the images in a many-to-many fashion according the the epipolar constraint. For each match a corresponding world position can be calculated using triangulation. Given that the dimensions of the object being searched for are known, using the world position, the *expected image* of the object in both cameras can be found. From the expected image we can calculate an *expected object*. The expected object is compared against the *observed object* found from segmentation. The object pair with the best comparison is said to be the object.

The *observed object* refers to the objects returned from mono-vision processing. The *expected image* refers to the image found by the stereo recognition system. It is what an object should look like at a certain world position. The *expected object* is what the expected image would look like if it underwent mono-vision processing. The method for comparing the observed object and the expected object will depend on the mono-vision processing used. A particular implementation is described in section 3.

There is a limit to this methodology. Knowing the position of an object is generally not enough to determine it's expected image: it's rotation must also be known. The single exception to this rule is a sphere, which is rotationally independent. The implementation described in section 3 is for spheres only. Extending the methodology to arbitrary objects requires particular points on an object to be located. For example, when finding a cube, it would be necessary to locate some of it's corners. Approximate methods to deal with non-spherical objects are discussed in section 5.

2.1 Matching

The objects are matched according to the epipolar constraint. The matching is many-to-many. Referring to Figure 2, there are three objects A, B and C. Their images are A_0, B_0 and C_0 in camera 0 and A_1, B_1 and C_1 in camera 1. The epipolar lines are the dashed lines. A and B lie on the same epipolar lines and will be matched together. The matches are $A_0 \Leftrightarrow A_1$, $A_0 \Leftrightarrow B_1$, $B_0 \Leftrightarrow A_1$, $B_0 \Leftrightarrow B_1$, $C_0 \Leftrightarrow C_1$. Note that $A_0 \Leftrightarrow B_1$ and $B_0 \Leftrightarrow A_1$ have matched different objects together. These are incorrect matches. They will be rejected in later stages.

This is the most important notion of the methodology and the fundamental result of this paper. There is no need to enforce that objects are matched one-to-one at this stage. To fortify this point we state that for an

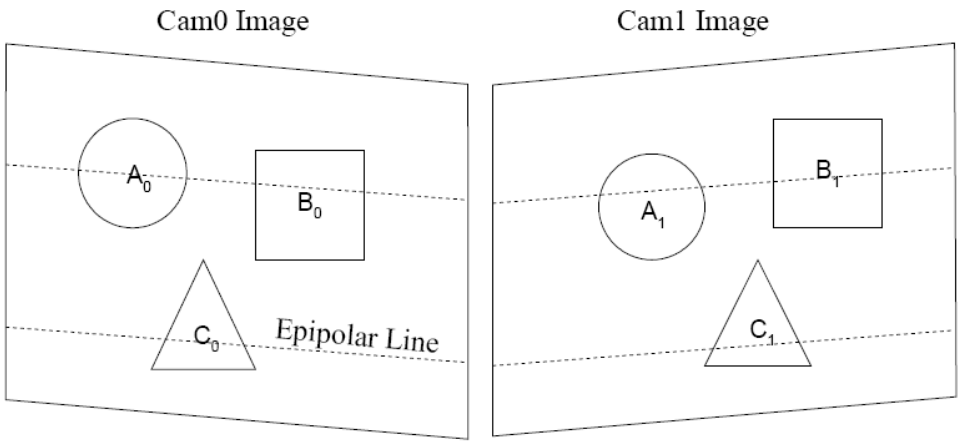

Figure 2: Matching the blobs

incorrect match, the world position calculated from this match will not point to an existing object. If the image of this non-existent object is calculated, it should not be similar to the image of the matched objects, the observed images. In fact, it is highly likely that the world position calculated from the match would not even be a position visible in the cameras.

2.2 Calculating the Expected Image

Given a match, the world position of an object related to the match can be calculated by triangulation. Given that the dimensions of the object are known, the image of the object in both cameras can be calculated. The expected image is calculated as follows:

Let S_w be the set of Cartesian points that represent the surface of the object when it is centered on the origin. Let R be the rotation of the object, T be the translation (world position) and P be the camera matrix. Then the expected image of the object is denoted by the set of points S_i,

$$S_i = PTRS_w$$

There are some problems with this expression. The rotation matrices R and T must be calculated. T is calculated using stereo-vision and triangulation. As discussed, the calculation of R is not performed. Another difficulty is that the elements in S_i are not unique. The implementation described in section 3 avoids uniqueness in S_i issues by choosing a particular type of mono-vision image segmentation. It may be that other implementations, need to further consider uniqueness in S_i.

3 Implementation

The methodology was implemented for the soccer playing humanoid robot, GuRoo [Wyeth *et al.*, 2001]. The

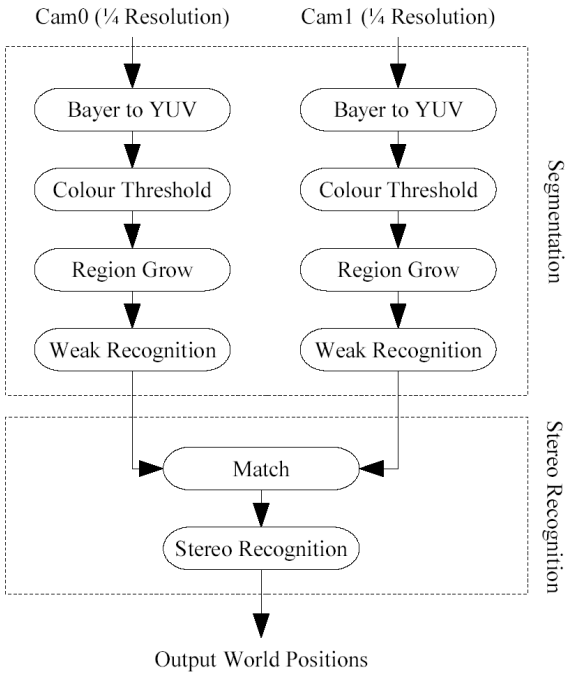

Figure 3: Block diagram of the humanoid implementation

objects to be found were a ball (sphere), poles (cylinder) and a goal (rectangle). The processor used is a low power VIA C3. Two cameras are mounted on the robots head roughly 1.3 meters from the ground. These cameras are used at a resolution of 123 by 164 pixels.

The cameras were calibrated using an implementation of Tsai's [1987] camera calibration technique using a calibration mat visible in Figure 12. The colours for each object are entered by a GUI using the mouse. This process is called colour calibration and is similar to the method described by Ball [2004].

3.1 Colour Thresholding and Region Growing

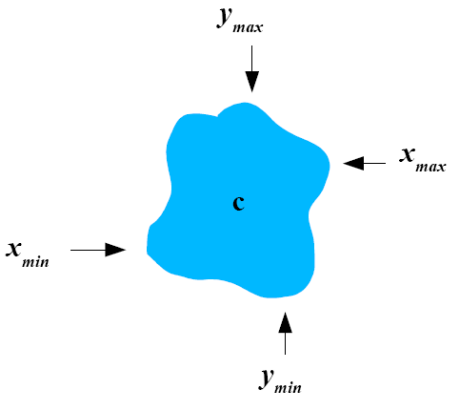

Figure 4: Region features

Prior to stereo-recognition, each image is undergoes colour thresholding and region growing. The regions found indicate possible objects. Important features extracted from each region are the *extrema*: x_{min}, x_{max}, y_{min} and y_{max} and the region centroid (Figure 4). We introduce further notation: the *expected extrema* refer to the extrema calculated from the expected images; the *observed extrema* are the extrema seen on the observed region.

3.2 Locating Spheres

The parametric representation of a sphere is used to represent the sphere at any calculated world position. This is:

$$S_w = \begin{pmatrix} -b\cos(v)\sin(u) \\ b\sin(v) \\ b\cos(v)\cos(u) \end{pmatrix}$$

where b is the ball radius and $u \in [0, 2\pi)$, $v \in [0, \pi)$. The sphere has the convenient property of rotational independence. We can calculate S_i without rotation R,

$$S_i = PTS_w$$

Written in full this gives,

$$S_i = \frac{\lambda}{b\cos(v)\cos(u) + t_z} \begin{pmatrix} -b\cos(v)\sin(u) + t_x \\ b\sin(v) + t_y \end{pmatrix}$$

where λ is the camera's focal length and the translation matrix T corresponds to a translation by (t_x, t_y, t_z).

We now wish to locate the extrema (Figure 4) within S_i. This is, with respect to u and v we wish to maximize and minimize:

$$\lambda \frac{-b\cos(v)\sin(u) + t_x}{b\cos(v)\cos(u) + t_z}$$

to find the x extrema, x_{max} and x_{min}; and

$$\lambda \frac{b\sin(v) + t_y}{b\cos(v)\cos(u) + t_z}$$

to find the y extrema, y_{max} and y_{min}

This is an optimisation problem. In fact, there exists an analytic solution for the extrema. However, it is unstable. Instead, it can be shown that the x extrema lie on the ring $v = 0$ and the y extrema lie on the ring $u = 0$. This is depicted in Figure 5. An incremental search is performed around these rings to locate the expected extrema for the ball.

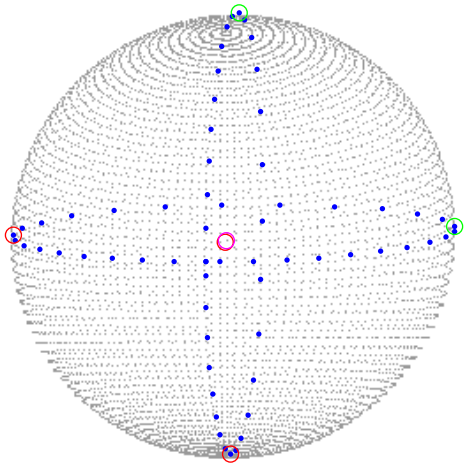

Figure 5: Searching rings for the extrema. x_{max} *and* y_{max} *extrema are marked with green circles.* x_{min} *and* y_{min} *extrema are marked with red circles. The centroid is the magenta circle.*

3.3 Error Calculation

A measure of error between expected and observed extrema is calculated by projecting the observed extrema

out from the camera to a point closest to the expected extrema. The distance between this point and the expected extrema is the error. The error has units in millimeters. The total error for a match is the sum of the error calculated from the 4 extrema for each region.

4 Results

4.1 General Robustness Test

This test demonstrates that the system can accurately recognise objects based on size and shape. A red shirt, red book, red ping pong bat and the ball are placed in the scene (Figure 6). Colour calibration is performed so that all objects are fully segmented. In this way the system cannot rely on colour to differentiate between the objects. The ball is moved around the scene and the ability of the system to recognize the ball over the other objects is observed. The error of each object is given in Table 4.1. The average error of the ball as it moves backwards in the scene is recorded in Table 4.1.

Only if the ball is placed ≈4m away does the recognition system fail. In this case the system finds the ping-pong bat. It is expected that error will increase as object distance increases. This is likely due to camera calibration error.

Object	*Error (mm)*
Ping-Pong Bat	63.58
Shirt	94.67
Red Book	87.26

Table 1: Error for each Object

Ball Distance (m)	*Error (mm)*
0.5	33.82
1	48.92
2	48.29
3	62.47
4	73.32

Table 2: Error for Ball at Different Depths

4.2 Big Ball vs Small Ball

This tests the ability of the system to distinguish between a ball of radius 90mm and a ball of radius 110mm. The colour calibration is set so that both balls are fully segmented. Sample images of the camera and region views are presented in Figure 8. The radius that the system is searching for is adjusted and the ball found by the system is observed. The search radius is set to 110mm, 100mm, 90mm 75mm and 60mm. The tests are conducted at depths of approximately 500mm, 1000mm,

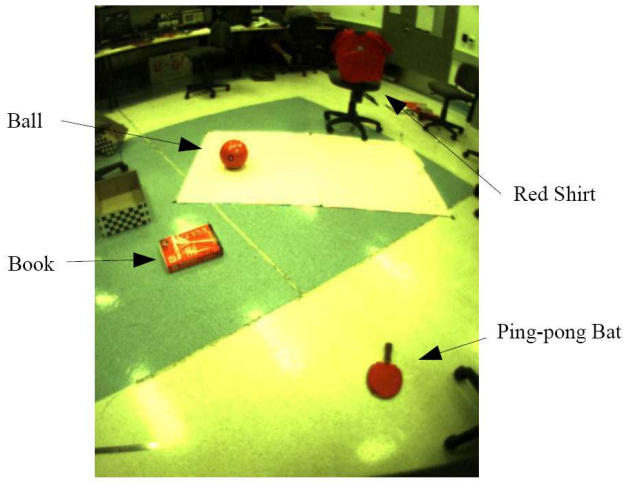

Figure 6: Ball ≈3m away

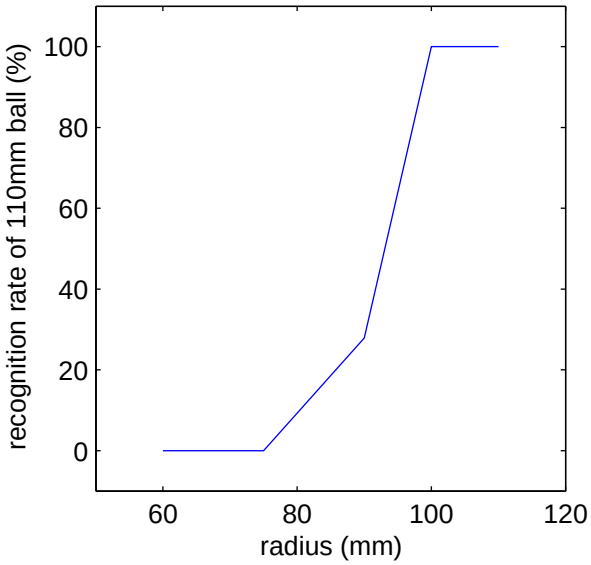

Figure 7: Recognition Rate of a 110mm ball vs 90mm ball as specified radius varies

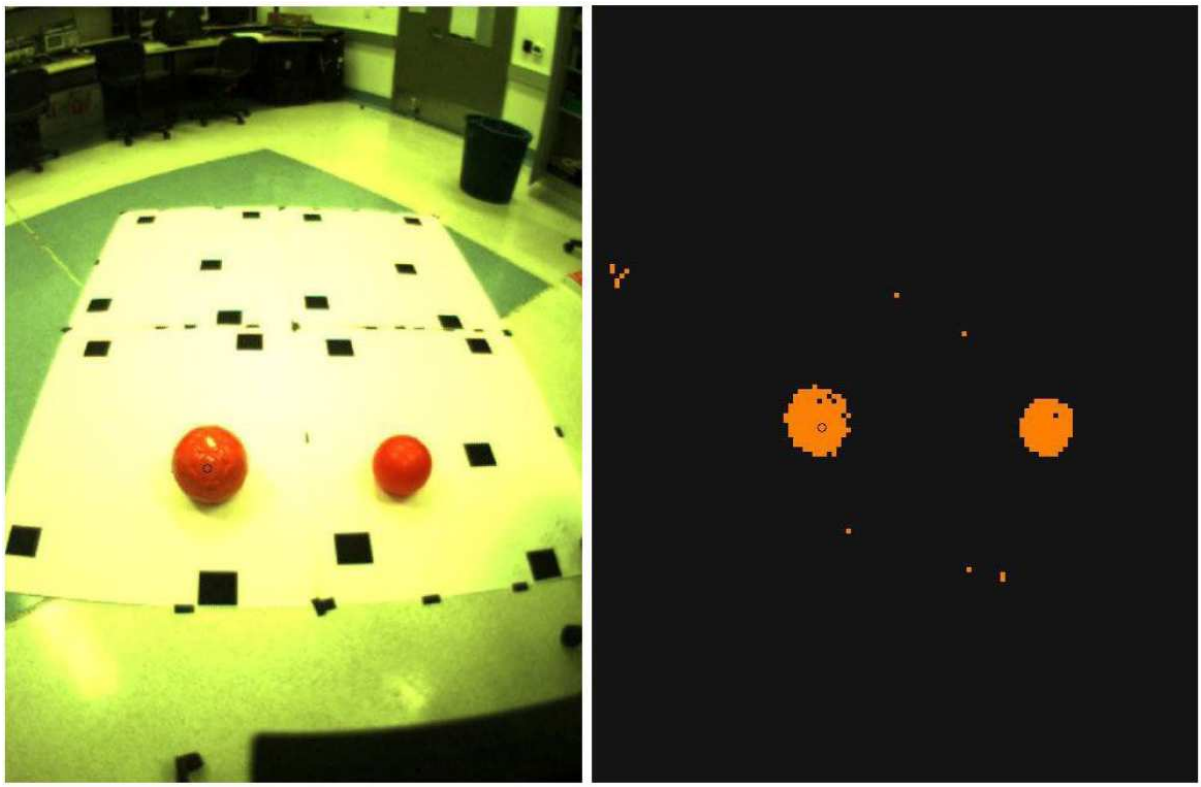

Figure 8: Picture of the radius test. *Right*: Camera0 Picture. *Left*: Camera0 Region View.

2000m and 3000mm. Figure 7 plots the recognition rate of the 110mm ball as the search radius increases.

The system begins to recognise the 110mm ball when the system radius is set to 90mm. The system recognises only the 110mm ball when the radius is set to 100mm. It would be expected that this shift should occur at 100mm, and not 90mm. It appears that the system is overestimating the size of the balls in the scene by approximately 10mm.

The most likely cause of this error would be if the system is underestimating the depth of the ball. The stereo recognition system would expect the ball to be larger than it is. This hypothesis is supported by the position accuracy tests.

4.3 Position Accuracy

The ball is placed at a number of measured locations in the scene along the floor. The magnitude and direction of the error in the x,z plane is plotted in Figure 9.

A significant result of the test is that the majority of error is an underestimation of depth, this can be seen as the large underestimation in the z coordinate in Figure 9. The cause of this error is suspected to be poor camera calibration and image noise causing the region centroids to move. This causes incorrect world positions to be calculated by the stereo system. The resolution of the images is only 123 by 164 pixels. Pixel quantisation would also cause position error.

It can be seen that the world position error decreases as the ball gets closer to the camera. This is to be expected as the effects of camera calibration error and limited resolution would increase with depth.

4.4 Invariance to Lighting Changes

The previous object recognition results show that the system is able to recognise objects based on size and shape with little reliance on colour segmentation. This gives the system an ability to handle dynamic lighting. A very wide range of colours can be set in colour calibration without degrading recognition performance so that the system can handle changing light.

To alter lighting in a controlled way, the camera's internal gain is adjusted in the range 0 to 195. The recognition rate and object recognition error are recorded. The system is set to search for a ball of radius 110mm. The 110mm ball, the 90mm ball and a red shirt are placed in the center of view. Figure 11 indicates what the image looks like for the different gain values.

The results show that the system can handle dynamic lighting conditions. These results test only global luminance changes in lighting conditions. It is suspected that the system can be colour calibrated to handle chromatic

and local changes in light with similar results.

4.5 Computational Performance

The VIA C3 CPU used, is roughly comparable to a PentiumIII at 400MHz. To test efficiency, the system was set to find all objects as pictured in Figure 12. The recognition rate for all the objects was 100%. The average frame rate during the test was 37.47fps. The average CPU usage was 81%. The time analysis for separate sections of the vision system is presented Table 3.

Vision Stage	*time %*
Image acquisition and YUV conversion	27.29%
Colour Thresholding	22.63%
Region Growing and Weak Recognition	29.32%
Ball Stereo Object Recognition	0.44%
Yellow Pole Stereo Object Recognition	0.48%
Purple Pole Stereo Object Recognition	4.02%
Goal Object Recognition	0.01%

Table 3: Time spent in each stage of the vision system

5 More that Just Spheres

This section describes how the humanoid system was modified to locate cylinders and rectangles. This involves making assumptions about the rotation of the object that would not be appropriate for all applications.

5.1 Locating Cylinders

The parametric representation of a cylinder is used to calculate the expected image of a cylinder. The surface of the cylinder is given by:

$$S_w = \begin{pmatrix} -r\sin(u) \\ v \\ r\cos(u) \end{pmatrix}$$

where r denotes the radius of the cylinder, $v \in [height/2, -height/2]$ and $u \in [0, 2\pi)$. The pole is rotationally dependent around axes other than its vertical axis. For the expected image to be calculated a method for obtaining the rotation matrix R is required.

Rather than attempt to calculate R from the image, the assumption is made that a pole is always standing upright. For the humanoid this is a reasonable assumption; all poles of importance are upright and it is desired that a pole lying flat not be identified. Given this, the rotation of a pole is acquired by the camera's rotation with respect to the ground plane. This rotation is calculated during camera calibration.

Given rotation R the expected image, S_i is calculated by the formula in section 2.2. Optimisation techniques can be used to locate the expected extrema. Due to the existence of a rotation R, the problem is more complex

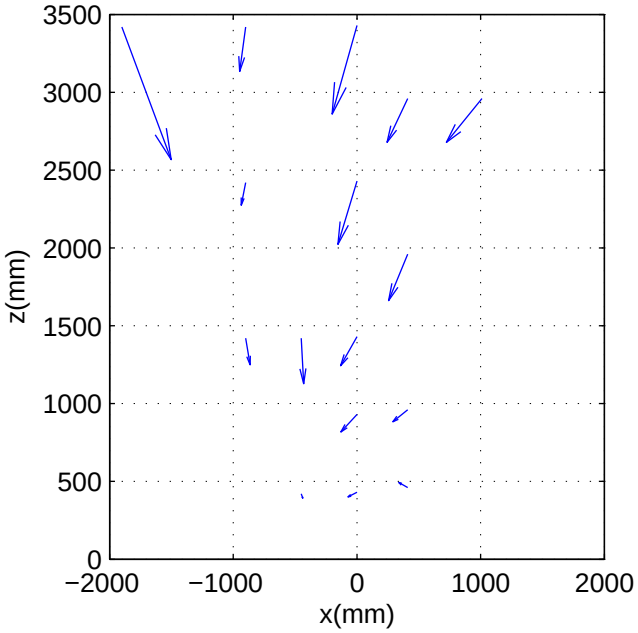

Figure 9: Direction and magnitude of world position error in the x,z plane

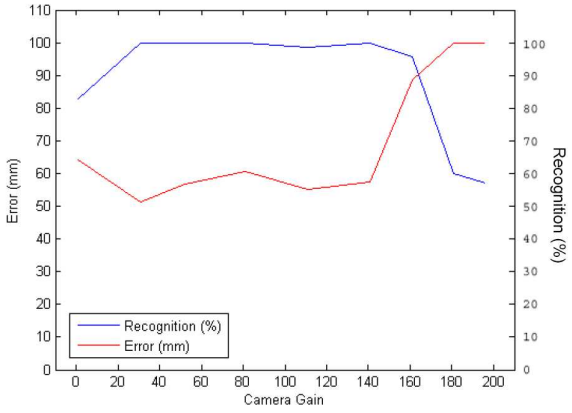

Figure 10: Ball recognition as lighting conditions change

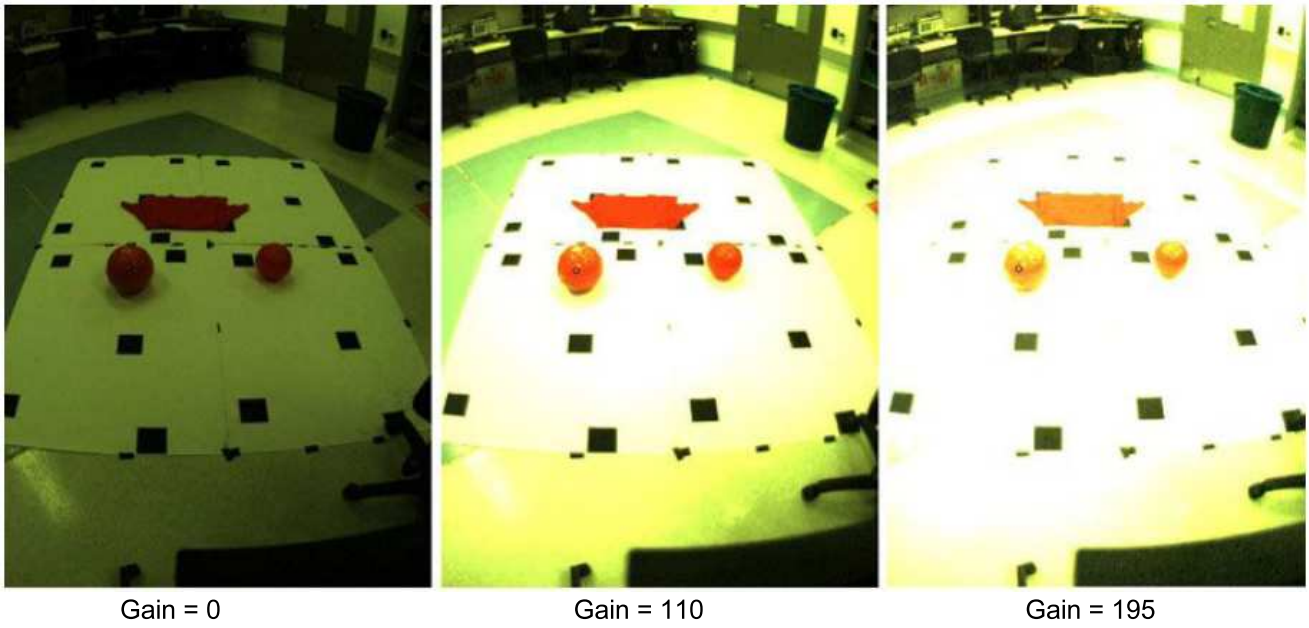

Figure 11: Camera0 images of different lighting conditions

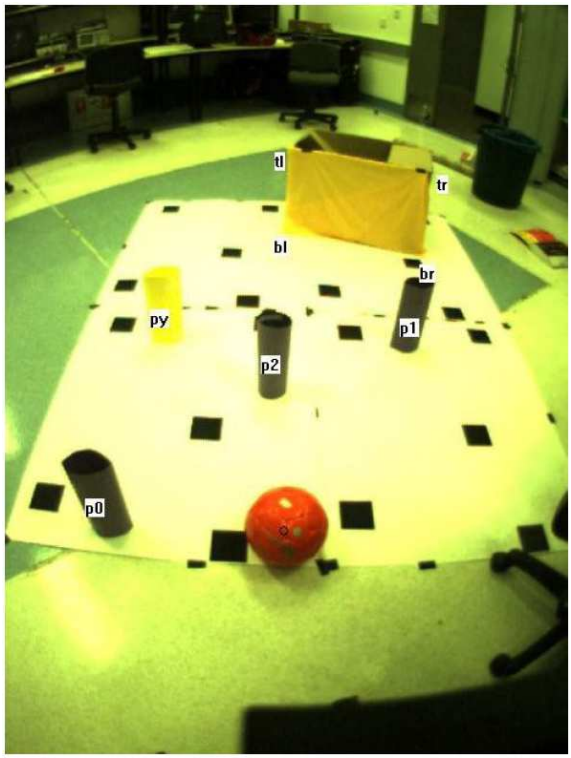

Figure 12: Picture with all objects located. This setup was used for the computational test.

than is was for the ball. To simplify the problem, it is helpful to first realise that all extrema occur on the rings $v = goalheight/2$ and $v = -goalheight/2$. This is displayed in Figure 13. That extrema occur on the rings is attributed to the Extreme Value Theorem.

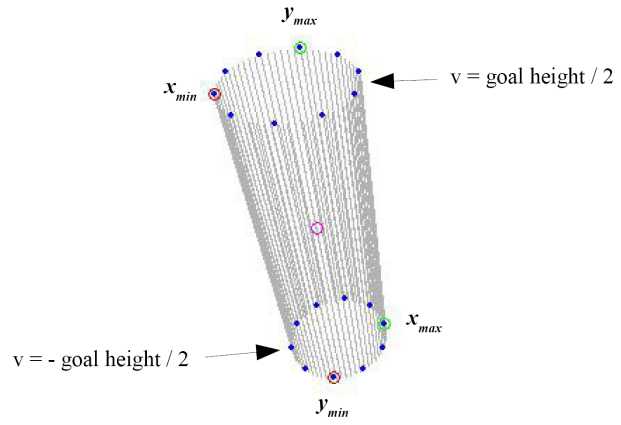

Figure 13: Cylinder extrema. *The extrema lie on rings on the top and bottom of the cylinder*

As with the sphere, a search is performed around the rings to find the extrema. Note that with the sphere, it was certain that all y extrema lay on one ring ($u = 0$) and all x extrema lay on another ring ($v = 0$). This is not the case with the cylinder. It is possible for some or all of the extrema to lie on one of the rings only. The

search procedure must check for this.

Once the extrema for all cylinder matches have been found, the error of each match is calculated according to section 3.3. The match with the least error is said to be the cylinder of interest. Figure 12 shows 3 purple poles being located. In this case, the best 3 matches are accepted.

5.2 Locating Rectangles

The humanoid was required to find a goal, represented by a yellow rectangle. A rectangle is represented by it's corners. Unlike, the sphere and cylinder, where only the centroid of the object was important, all 4 rectangle corners need to be located. Also unlike the sphere and cylinder, the rectangle is rotationally dependent around all axes. As for the cylinder, the goal is always assumed to be standing upright. The approximation of the rectangles pitch and tilt is taken from the camera calibration. The yaw of the rectangle is still unknown. The system must be able to obtain or approximate the yaw of the goal from the image itself.

To approximate the yaw the world position of the left and right edge of the goal is approximated. The center of the left and right bounding box edges of the rectangle is found in both cameras. The world position corresponding to these points is calculated using triangulation. Given that the world position of the center of the left and right edges is (x_r, y_r, z_r) and (x_l, y_l, z_l) respectively. An approximation to the yaw is given by,

$$yaw \approx \arctan\left(\frac{z_r - z_l}{x_r - x_l}\right)$$

Once the position and rotation of the rectangle is known, it's expected image, S_i can be calculated (section 2.2). S_i contains the expected image positions of the goal corners. The observed bounding box of the goal is known. The expected bounding box is calculated from S_i. The error measurement is calculated by comparing the observed and expected bounding boxes. The match with least error is said to be the rectangle.

6 Discussion

The paper presents a fast and robust stereo object recognition method. The method is currently unable to identify the rotation of objects. This makes it very good at locating spheres which are rotationally independent. Other objects can be located provided that assumptions are made for there rotation.

Fundamental to the method is that the correspondence problem is solved using information about the dimensions of the object. Possible objects are matched in a many-to-many fashion according the the epipolar constraint, the object is then found by comparing the observed images with the expected images calculated from the object matches. This method is in contrast to previous stereo object recognition systems where the scene is first reconstructed by point matching techniques. The object would then be located by finding where it best fits into the reconstruction.

The main advantages of the method are:

- Suitable for real-time application on embedded devices
- The recognition is significantly robust against image noise and poor camera calibration.
- The system uses colour.

There are two current weaknesses:

- There is no obvious way to obtain the rotation of objects. This currently limits the system to finding spheres. For some applications, approximations for object rotation can be made to find simple objects.
- The system does yield object position with high precision.

The weaknesses currently limit the method to tasks that require rapid, efficient and consistent location of simple objects in noisy conditions without very high position accuracy. These properties make the method particularly suitable for robot soccer.

References

[Ball *et al.*, 2004] Ball, Wyeth and Nuske. A Global Vision System for a Robot Soccer Team In *Proceedings of the 2004 Australasian Conference on Robotics and Automation (ACRA)*, Canberra, Australia, 2004.

[Dijck and Heijden, 2002] Dijck, H. and Heijden, F. Object recognition with stereo vision and geometric hashing. In *Pattern Recognition Letters* 24, pp. 137146, 2003.

[Faugeras, 1992] Faugeras, O. What can be seen in three dimensions with an uncalibrated stereo rig? In *Proceedings ECCV 92, Lecture Notes in Computer Science*, 588, Springer-Verlag, 1992.

[Hartley and Zisserman, 2004] Hartley, R. I. and Zisserman, A. Multiple View Geometry in Computer Vision. 2nd Edition, 2004, Cambridge University Press, ISBN: 0521540518.

[Leung *et al.*, 2004] Leung, Carlos and Appleton, Ben and Lovell, Brian C. and Sun, Changming. An Energy Minimisation Approach to Stereo-Temporal Dense Reconstruction. In *International Conference on Pattern Recognition*, vol. 4, pp. 72-75, Cambridge, United Kingdom, 23-26 August, 2004.

[Ruzon and Tomasi, 1999] Ruzon, M. and Tomasi, C. Color Edge Detection with the Compass Operator. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, vol 2, pp. 160-166, June 1999.

[Rygol *et al.*, 1991] Rygol, M., Pollard, S., and Brown, C Multiprocessor 3D vision system for pick and place In *Image and Vision Computing*, 9(1):3338, 1991.

[Sumi *et al.*, 2002] Sumi, Kawai, Yoshimi and Tomita. 3D Object Recognition in Cluttered Environments by Segment-Based Stereo Vision. In *International Journal of Computer Vision* 46(1), pp. 5-23, 2002.

[Tsai, 1987] Tsai, A. Versatile Camera Calibration Technique for High-Accuracy 3D Machine Vision Metrology Using Off-the-Shelf TV Cameras and Lenses. In *IEEE Journal of Robotics and Automation*, 1987.

[Wolfson and Rigoutsos, 1997] Wolfson, H. and Rigoutsos, I. Geometric Hashing: An Overview. In *IEEE Computational Science and Engineering* October-December 1997.

[Wyeth *et al.*, 2001] Wyeth G., Kee, D. and Wagstaff, M. Design of an Autonomous Humanoid Robot *Proceedings of the Australian Conference on Robotics and Automation*, November 14-15, Sydney 2001