Combining Edge Detection and Colour Segmentation in the Four-Legged League

Craig L. Murch and Stephan K. Chalup*
School of Electrical Engineering & Computer Science
The University of Newcastle, Callaghan 2308, Australia
{cmurch,chalup}@eecs.newcastle.edu.au

Abstract

Humans process images with apparent ease, quickly filtering out useless information and identifying objects based on their shape and colour. In the Four-Legged League of RoboCup, the focus of most vision systems has been on using colour to recognise objects. This single-mindedness has many disadvantages, but edge detection and shape recognition are not panaceas as their computational requirements are too high. This work focuses on a technique to efficiently combine the use of colour and edge detection in order to better recognise the field landmarks and objects found in the controlled environment of the Four-Legged League of RoboCup.

1 Introduction

Robot vision systems are often required to identify landmarks relevant to the operation of the robot. In some cases, colour alone can be used to identify landmarks. For other objects and landmarks, edge detection and shape recognition techniques can be used.

The Four-Legged League [Four-Legged League, 2004] of RoboCup [RoboCup, 2004], is concerned with programming fixed hardware - specifically, the Sony AIBO [AIBO, 2004] - to play soccer on a specially designed field. Colour is currently the only criterion used to identify landmarks and objects on the soccer field, making colour classification a critically important part of the vision system. Generally speaking, colour classification is performed by using a pre-computed table (a 'colour table') [Bruce *et al.*, 2000; Quek, 2000] which maps raw colour information into a small set of colours of interest to the robot. Colours of interest to the robot are often termed 'classified colours'. Typically, clumps of classified colour in the image are formed into 'blobs', a process referred to as blob formation. These steps are illustrated in figure 1. Blobs are then processed by various ad-hoc techniques to determine which objects appear in the image. A number of methods to perform automatic generation [Cameron and Barnes, 2004; Mayer *et al.*, 2002] and modification [Jüngel *et al.*, 2004] of colour tables (and thus improve the accuracy of colour classification) have been developed.

Some teams in the Four-Legged League of RoboCup have developed efficient edge detection related methods to locate field boundaries [Burkhard *et al.*, 2002; Röfer *et al.*, 2003], particular points on the field [Seysener *et al.*, 2004], or even the ball [Treptow *et al.*, 2003]. While these methods can assist in localization, they mostly ignore the coloured landmarks around the field. Furthermore, they tend to be less robust and less accurate than the colour based methods.

Algorithms to combine edge detection and colour segmentation in a single vision system have been implemented in general [Saber *et al.*, 1997], but such techniques have yet to be widely applied in the highly time-critical environment of the Four-Legged League of RoboCup. Some useful ideas (as well as very different approaches to the problem, focused on scan-lines and edge detection) have been presented in Jüngel *et al.* [2004] and Röfer *et al.* [2003]. The following paper details the study and implementation of a joint colour and edge detection method intended to overcome the problems inherent in present Four-Legged League vision algorithms.

2 Motivation

There are many problems attached to the reliance on colour for object recognition. The distance to objects on the field is often dependent on their size in the image - but this size can be very inaccurate if colour classification is poor. Sub-optimal colour classification is virtually unavoidable. In particular, colour classification tends to become less accurate as lighting conditions change during the RoboCup competition [Wang *et al.*, 2002] - and this is despite the artificial lighting structures placed above robot soccer fields. Indeed, lighting conditions can even change mid-game due to the transient audience. To make matters worse, there has been a move to reduce the reliance on artificial light structures at RoboCup, meaning lighting conditions are likely to become even more varied (and thus challenging) at future competitions.

*http://www.robots.newcastle.edu.au

Figure 1: The top figure is the original, unprocessed image obtained from the robot's camera. The middle figure is the same image after colour classification has been performed. The lowest figure shows coloured blobs that have been formed based on the colour classified image.

The automatic colour table generation techniques already mentioned [Cameron and Barnes, 2004; Mayer *et al.*, 2002] can assist with the problem of lighting conditions changing over relatively long periods. However, most methods are unable to cope with real-time lighting variations - such as those caused by a moving audience or sudden changes in external lighting conditions - since their computational requirements are too high. Additionally, robots may cast shadows on the ball or other objects, causing localized lighting variations.

It seems clear that the heavy reliance on colour classification in the Four-Legged League is significantly impairing the performance of teams. Edge detection techniques appear to be the logical alternative, and these are under investigation by a number of teams. Unfortunately, edge detection algorithms and object recognition systems based on them tend to be very computationally demanding. The vision systems in the Four-Legged League generally operate in real-time at 30 frames per second. Even relatively simple edge detection and shape recognition tasks are often far too computationally expensive for the very limited processors in the Sony AIBO robots. For example, the Hough transform [Hough, 1962; Gonzalez and Woods, 2002] can be used to identify circles or other shapes in an image: the caveat being that it is too slow for real-time operation on a Sony AIBO robot.

Any vision algorithm used on the robot must be - above all else - accurate. Mistakes made by the object recognition system can be catastrophic, resulting in the robot kicking the ball the wrong way or even not seeing the ball at all. It can to expected then that teams are often slightly conservative in their vision system implementations: in fact, this may be analogous to the conservative use of machine learning in the Four-Legged League [Chalup and Murch, 2004].

Thus, an heavy reliance on colour classification can be extremely detrimental to the performance of robots playing soccer. This work presents a computationally efficient method to use edge detection in order to reduce the reliance on good colour classification. In brief, the technique is able to use edge detection to improve the effective colour classification on any given image. It is also possible that in future, the algorithm can be adapted to measure the performance of different colour classification mechanisms. Such an evaluation function would greatly assist in the machine learning of colour classification techniques.

3 Methods

The general technique will first be presented using the orange ball as an example. The algorithm itself also works for many of the other objects found on the RoboCup Four-Legged League soccer field: an adaption of the method to assist with landmark recognition is also described.

3.1 Ball Recognition

To recognise the ball, the algorithm considers the largest orange 'blob' in the image. If the colour classification is very good, this blob will match almost exactly the size of the ball in the image. More realistically, the blob will encompass some subset of the ball in the image. The goal is to determine the real edges of the ball despite the poor colour classification.

Let us consider the projection of *n* rays outwards from the centre of the blob. Along each ray, we perform edge detection on the raw image until a sharp edge is found. An heuristic based on the colour classified image is used to assist the search: if several pixels examined by the algorithm were classified as colours other than orange, then a pixel identified as orange must be found soon after or the search will end. Too, if the edge of the image is encountered when searching for the edge, the search is abandoned. In this way, the method is able to make use of available pixel colour information to improve its edge detection performance. Note that this heuristic can have undesirable consequences in cases of images with lots of misclassified pixels.

Where an edge is detected, we have the edge of the ball in the image. Given the coordinates of these edges, we can now use a circle fitting algorithm [Seysener *et al.*, 2004] to find the actual centre and radius of the ball. It is possible to use other methods (e.g., just three points are required to completely describe a circle), but circle fitting methods tend to be more robust in noisy conditions. Before circle fitting, however, the outlier rejection technique described in section 3.3 is executed. If the circle fit found is sufficiently good (as determined by a standard deviation, which is returned as part of the circle fitting algorithm), then the ball has been found and its correct size in the image identified. The exact 'goodness' of fit required for the ball to be detected is set based on empirical data and can also be influenced by certain other sanity checks (e.g., if the ball is not at the expected elevation, the fit must be better).

3.2 Landmark Recognition

The method is easily adaptable to the field landmarks, as they too are simple shapes (rectangles). The Four-Legged League soccer field has a number of coloured 'beacons' placed at its corners. Each beacon consists of pink and one other colour; Yellow or blue. These are usually identified by looking for blobs of the appropriate colours one over the other, with the distance (in pixels) between the centre of each coloured blob used to determine the distance. However, the blobs found are often not in the centre of the beacon colours. Further, if a blob is too small it will not be considered to be part of a beacon (this is a simple noise filtering method).

The soccer field also has coloured goals at either end of it. The goal colours are yellow and blue, and appear as a large solid rectangle to the robot. Again, these are identified by finding appropriately sized blobs of the correct colour.

To adapt the algorithm to goals and beacons, we need only be able to enlarge the blobs that are used to recognise them. Once again, we project rays outwards and perform edge de-

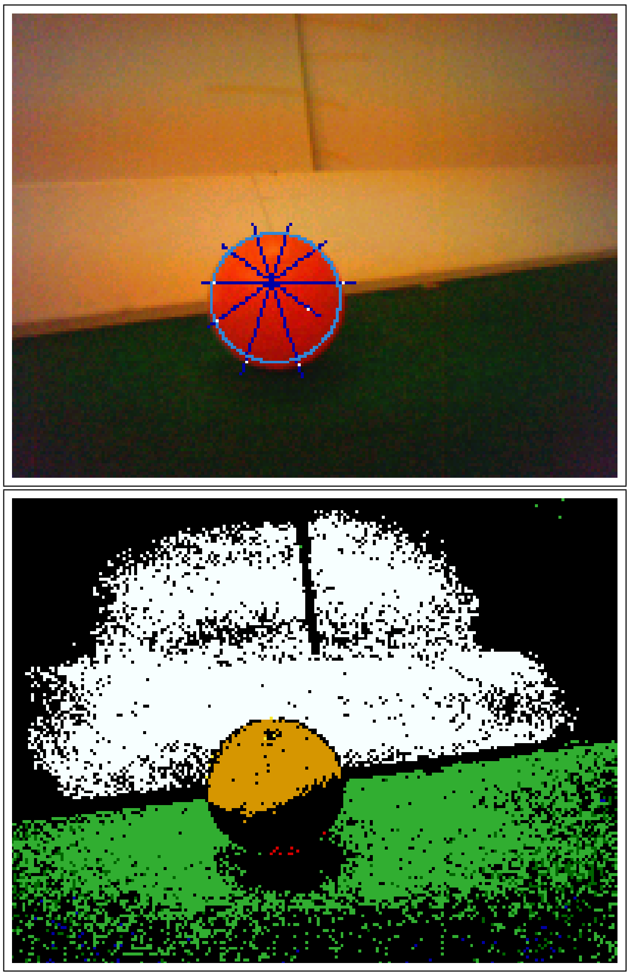

Figure 2: Edge detection rays are shown in dark blue on the unclassified image at the top. Detected edge points are shown in white, and the fitted circle is in light blue around the ball. The poorly classified image is shown on the bottom: notice the fitted circle is close to the real shape of the ball despite the poor colour classification.

tection. This gives us a set of points which should correspond to the extreme points of the coloured object we are examining. At this point, we perform the outlier rejection technique described in section 3.3. We then enlarge the blob so that it encompasses these extreme points. In this way, blobs that are too small due to poorly classified colour should be automatically expanded to their correct size.

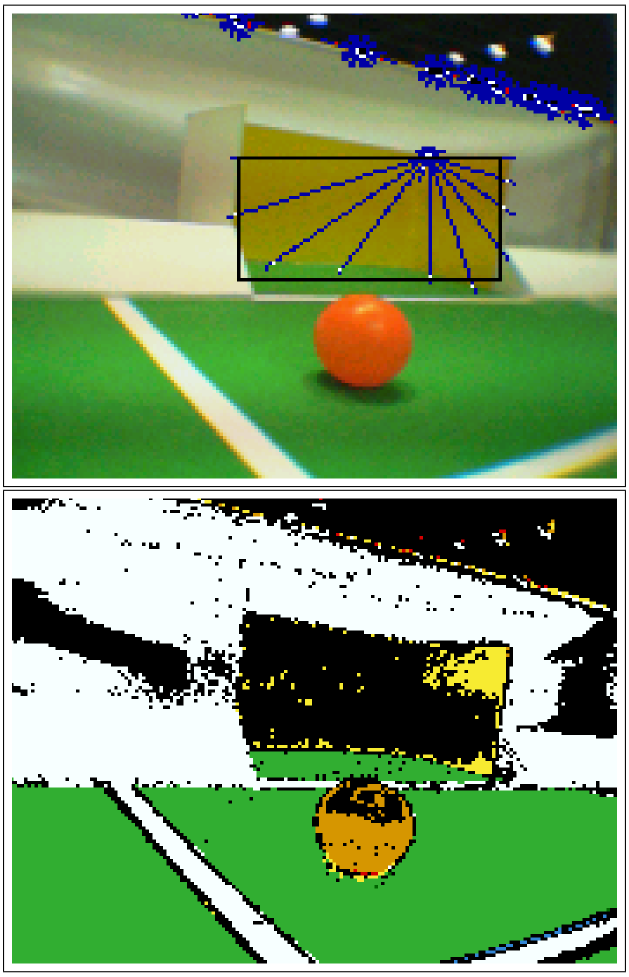

Figure 3: Edge detection rays are shown in dark blue on the unclassified image at the top. Detected edge points are shown in white, and the blob formed around the edge points is shown in black. The poorly classified image is shown at the bottom: notice the detected blob is close to the real shape of the goal despite this. Also note that an attempt has been made to extend the yellow pixels at the top of the image, but this has not been a problem.

3.3 Outlier Rejection

There are cases in which the algorithm detects an edge which it should not have. For example, white or yellow reflections from the lights can appear towards the top of the ball in the image, causing edges to be found prematurely.

The outlier rejection method chosen is a straightforward technique based on statistics, and the same general method is used for both landmark and ball detection. The intuitive idea is that the mean of all the edge points should be roughly equidistant from all the edge points, due to the relatively symmetric shapes of the objects that must be detected. Any edge point without this property is likely an outlier where either an edge was found too early or too late.

More formally, the mean coordinate is defined as the average position in the image of all the edge points found. The distance of the edge points from this mean coordinate is calculated. The variance of these distance values is efficiently calculated using equation 1 where n is the number of edge points found and $\{y_1,y_2,...,y_n\}$ are the distances of the edge points from the mean coordinate.

$$\sigma^2 = \frac{1}{(n-1)}(\sum_{i=1}^{n} y_i^2) - \frac{(\sum_{i=1}^{n} y_i)^2}{(n-1)n} \tag{1}$$

The sensitivity of the outlier rejection algorithm must be determined empirically. In these experiments, any edge point with a distance from the mean of more than two standard deviations (2σ) is rejected as an outlier. In this way, edge points that are inconsistently located compared to the majority are removed.

3.4 Edge Detection Techniques

The algorithm described in this work is reliant on the availability of a robust and reliable edge detection technique. There is an enormous number of edge detection techniques available, each intended to be sensitive to particular types of edges [Gonzalez and Woods, 2002]. Some effort has been expended testing and tuning different edge detectors (specifically, variations based on Laplace operators), but no alternative algorithms tested so far have consistently outperformed the relatively simplistic method described below. This is a surprising outcome, but may be explainable by the motion blur that is often present in images captured by the robot's rapidly moving camera. Specifically, the transient blur induced by the camera's movement tends to make the task of choosing edge detection threshold values very difficult. Additionally, the Sony AIBO camera is of poor quality and tends to produce noisy images. It is possible that a more in depth search would locate a better performing edge detector.

The Sony AIBO's camera supplies image information in the YUV colour space. Roughly speaking, the Y component (or 'channel') of a particular pixel corresponds to its intensity level, while the U and V components describe the pixel's colour. The currently implemented edge detection technique is as follows: if the difference in the Y channel of two adjacent pixels located along a ray is of sufficient size, an edge is detected. The magnitude of the difference in Y channel values is determined by some threshold value. A small extension has been implemented so that non-adjacent pixels are also able to define an edge.

A rather novel variation on the above technique has also been tested. It is based on the observation that edges tend to occur quite close to the detected 'end' of a projected ray (this end point is detected by a heuristic that checks for the the presence of several white or green pixels). Classification of the white and green pixels which this end point determination relies on tends to degrade relatively slowly under varying lighting conditions. This leads to a modification of the existing technique whereby the edge detection threshold is decreased (meaning it is easier to find edges), but the edge detection algorithm now starts from the 'end' of each projected ray rather than the middle of the object. Essentially, it is possible to increase the sensitivity of the edge detection algorithm because we know we are very close to the actual edge already. While this variation improves performance in general, it does not seem to perform well on the ball and so is disabled for the colour orange.

4 Results

The goal of the method employed is to improve the effective colour classification on individual images. Therefore, we must have a baseline corresponding to 'perfect' colour classification in order to obtain useful results. Fortunately, the time consuming task of creating a near optimal colour table for each image in a reasonable size test set has already been performed in Quinlan *et al.* [2003]. It is this near optimal colour classification that we will use to evaluate the algorithm's performance.

The algorithm described does not modify the actual colour classified image, as it operates on blobs associated with possible field objects rather than on the classified image directly. This means we cannot directly compare colour classification results. Instead, we construct 'blob images' by drawing all of the blobs that form objects identified by the vision system (one blob image per original image). This process is shown in figure 4. We can then examine the differences between the blobs formed to approximate the performance of colour classification. In a sense, we are comparing the performance of the technique at a level slightly above simple colour classification. Note that the algorithm has been simplified for easier benchmarking, in that we now enlarge the ball blob as well as identifying a more accurate edge around it.

The test set consists of 60 images taken at a competition field. The 'initial' results are those of the original vision system with no enhancements. The 'with edge detection' results are obtained when the enhancement algorithm is in use, while the 'with edge detection (and backtracking)' is also using the enhancement algorithm except with a modified edge detection method (see section 3.4). The same colour table is used for all the above methods. The baseline results are obtained by using the 'initial' method with a near optimal colour table for each image: for convenience, we will term the blob images generated this way as 'near optimal'. An error is defined as any pixel that is classified differently from the near optimal

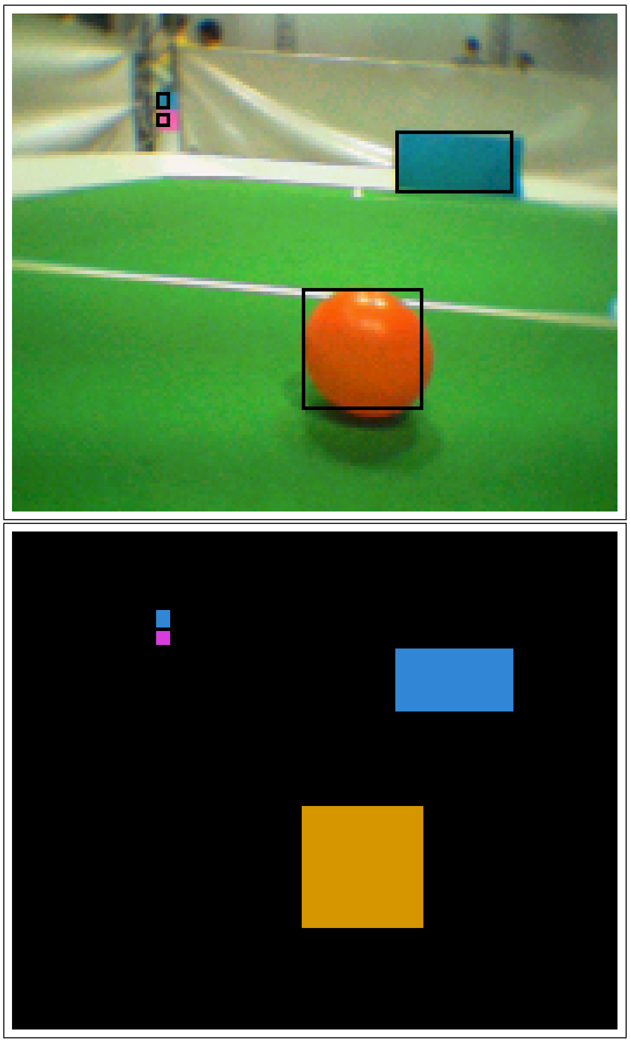

Figure 4: Conversion of a raw image (top) to a blob image (bottom). Blobs associated with vision objects found in the raw image are outlined in black. These blobs are then used to construct the blob image.

Colour	Initial Error %	With Edge Detection Error %	With Edge Detection (and backtracking) Error %
Orange	9.7 (6.5)	7.2 (4.4)	N/A
Blue	64.0 (16.7)	46.9 (24.4)	34.6 (12.6)
Pink	54.1 (11.2)	33.4 (17.9)	29.8 (16.8)
Yellow	17.8 (5.0)	14.1 (4.1)	14.9 (4.8)

Table 1: Classification error percentages resulting from the use of the enhancement algorithm. The values shown in brackets are standard deviations.

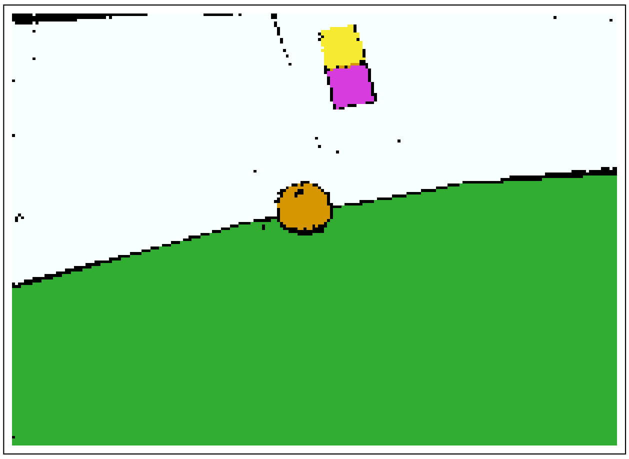

Figure 5: A classified image using a near optimal colour table. Notice the black outlines around the beacon and the ball, and also between green field and the white boundary.

blob image - that is, any deviation from the baseline result is deemed to be an error. Percentages shown in table 4 are the number of errors for that colour divided by the total number of pixels of that colour.

Note that there is an inherent issue with the use of near optimal colour tables in comparing algorithm performance. Images classified using near optimal colour tables tend to have an outline of 'unknown colour' (shown as black) around coloured blobs, corresponding to the edges of the blobs. Figure 5 is an example of this. These outlines are not considered to be part of blobs. However, the edge detection method locates these outlines as edges and includes them within their corresponding blobs.

The algorithm appears to provide a systematic improvement to the vision system when a poor colour table is in use. There is, however, a bad case which is not exposed in our test set. It must be remembered that this algorithm is running on a moving Sony AIBO robot: when the robot's head moves rapidly, the image can become blurred. This causes edge detection techniques to degrade, as soft edges are much harder to recognise. Figure 6 is an image captured by the robot while

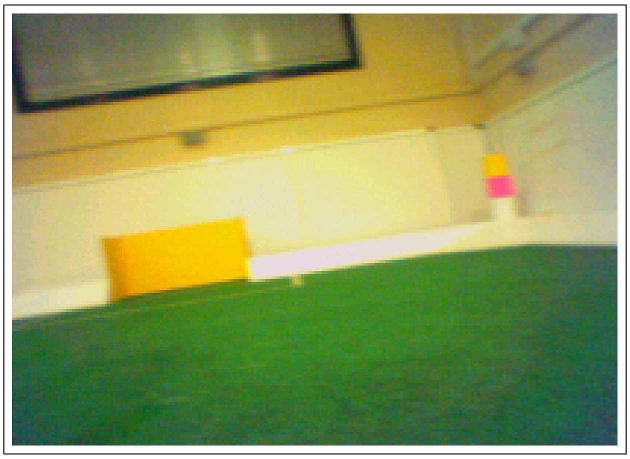

Figure 6: An image that has blurred due to the robot's head motion.

the head is moving at high speed: notice the softness of all the edges present in the image.

5 Discussion

The implementation of real time vision systems for use in robot soccer is a complex task, and includes many difficult problems. This study focussed on the development of an edge detection technique able to work in tandem with a more traditionally used colour segmentation approach to improve the resultant vision system. The constructed system can operate at the required 30 frames per second on a Sony AIBO robot.

This effective enhancement of colour segmentation allows measurements to landmarks can be more accurately determined even in times of poor classification. Similarly, a combination of colour segmentation, edge detection and efficient real-time circle fitting allows a similar improvement in the detection of the ball.

The main problem encountered with the algorithm is when the robot's head moves rapidly and the image becomes blurred. This negatively impacts the suitability of the technique for use in fast moving games of robot soccer, in which the robot's head tends to move rapidly almost constantly. It is not yet clear whether this issue can be addressed by modifying the edge detection thresholds based on the robot's motion. However, there is an application of the technique despite the image blur problem - in evaluating the performance of different colour tables. More precisely, it should be possible to measure the modification the algorithm's execution has performed: the smaller the modification, the closer the colour table tested is to optimal.

Due largely to the problem of image blur, the use of the algorithm in a real robot soccer game is perhaps some way off.

Acknowledgements. We would like to thank all past and present members and contributors to the University of Newcastle Four-Legged League RoboCup team.

References

[AIBO, 2004] AIBO. The Sony AIBO Entertainment Robot web site, 2004. http://www.aibo.com.

[Browning *et al.*, 2004] B. Browning, D. Polani, A. Bonarini, and K. Oshida, editors. *RoboCup-2003: Robot Soccer World Cup VII*, LNCS. Springer, 2004.

[Bruce *et al.*, 2000] J. Bruce, T. Balch, and M. Veloso. Fast and inexpensive color image segmentation for interactive robots. In *Proceedings of the 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS-2000)*, volume 3, pages 2061 – 2066, October 2000.

[Burkhard *et al.*, 2002] H.-D. Burkhard, U. Düffert, J. Hoffmann, M. Jüngel, M. Lötzsch, R. Brunn, M. Kallnik, N. Kuntze, M. Kunz, S. Petters, M. Risler, O. von Stryk, N. Koschmieder, T. Laue, T. Röfer, K. Spiess, A. Cesarz, I. Dahm, M. Hebbel, W. Nowak, and J. Ziegler. German Team - RoboCup 2002. Technical report, 2002.

[Cameron and Barnes, 2004] D. Cameron and N. Barnes. Knowledge-based autonomous dynamic colour calibration. In Browning et al. [2004].

[Chalup and Murch, 2004] S. K. Chalup and C. L. Murch. Machine learning in the four-legged league. In *3rd IFAC Symposium on Mechatronic Systems*. IFAC, 2004.

[Four-Legged League, 2004] Four-Legged League. Robocup four-legged league web site, 2004. http://www.openr.org/robocup/index.html.

[Gonzalez and Woods, 2002] Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing*. Prentice Hall, second edition, 2002.

[Hough, 1962] P. V. C. Hough. Methods and means for recognizing complex patterns. U.S. Patent 3,069,654, 1962.

[Jüngel *et al.*, 2004] M. Jüngel, J. Hoffmann, and M. Lötzsch. A real-time auto-adjusting vision system for robotic soccer. In Browning et al. [2004].

[Mayer *et al.*, 2002] G. Mayer, H. Utz, and G. K. Kraetzschmar. Towards autonomous vision self-calibration for soccer robots. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS-2002)*, volume 1, pages 214–219, September-October 2002.

[Quek, 2000] F. K. H. Quek. An algorithm for the rapid computation of boundaries of run-length encoded regions. *Pattern Recognition Journal*, 33(10):1637–1649, October 2000.

[Quinlan *et al.*, 2003] M. J. Quinlan, S. K. Chalup, and R. H. Middleton. Techniques for improving vision and locomotion on the AIBO robot. In *Australasian Conference on Robotics and Automation (ACRA-2003)*, 2003.

[RoboCup, 2004] RoboCup. RoboCup web site, 2004. http://www.robocup.org.

[Röfer *et al.*, 2003] T. Röfer, H.-D. Burkhard, U. Düffert, J. Hoffmann, D. Göhring, M. Jüngel, M. Lötzsch, O. von Stryk, R. Brunn, M. Kallnik, M. Kunz, S. Petters, M. Risler, and M. Stelzer. German Team - RoboCup 2003. Technical report, 2003.

[Saber *et al.*, 1997] E. Saber, A.M. Tekalp, and G. Bozdagi. Fusion of color and edge information for improved segmentation and edge linking. *Image and Vision Computing*, 15(10):769–780, October 1997.

[Seysener *et al.*, 2004] C. J. Seysener, C. L. Murch, and R. H. Middleton. Extensions to object recognition in the four-legged league. In D. Nardi, M. Riedmiller, and C. Sammut, editors, *Proceedings of the RoboCup 2004 Symposium*, LNCS. Springer, 2004.

[Treptow *et al.*, 2003] A. Treptow, A. Masselli, and A. Zell. Real-time object tracking for soccer-robots without color information. In *Proceedings of the European Conference on Mobile Robotics (ECMR 2003)*, 2003.

[Wang *et al.*, 2002] Z. Wang, J. Wong, T. Tam, B. Leung, M. S. Kim, J. Brooks, A. Chang, and N. Von Huben. rUNSWift UNSW RoboCup2002 Sony Legged League Team. Technical report, University of New South Wales, 2002.