Fast Object Detection with Foveated Imaging and Virtual Saccades on Resource Limited Robots

Adrian Ratter, David Claridge,
Jayen Ashar, and Bernhard Hengst

School of Computer Science and Engineering,
University of New South Wales, UNSW Sydney 2052 Australia

Abstract. This paper describes the use of foveated imaging and virtual saccades to identify visual objects using both colour and edge features. Vision processing is a resource hungry operation at the best of times. When the demands require real-time robust performance with a limited embedded processor, the challenge is significant. Our domain of application is the RoboCup Standard Platform League soccer competition using the Aldebaran Nao robot. We describe algorithms that use a combination of down-sampled colour images and high-resolution edge-detection to identify objects in varying lighting conditions. Optimised to run in real time on autonomous robots, these techniques can potentially be applied in other resource limited domains.

1 Introduction

Real-time identification of objects in a video feed is a significant research area in robotics, and forms the major component of many perception systems. For the rich environments we encounter in everyday life this is still an open research problem. RoboCup Soccer [5] is an international research and education initiative that constrains the environment to a soccer field with a limited number of objects, namely a ball, field, goal-posts, landmarks, and other robots. Vision algorithms are able to exploit these constraints, but face significant challenges.

Autonomous robots are limited in their processing power. Vision needs to share this limited resource with other functions such as world-modelling and behaviour generation. Success in soccer also depends on the speed at which robots can react. A major challenge is for the vision system to deliver real-time object recognition at maximum frame-rates and still leave resources for the other functions.

Colour cameras provide a high native pixel resolution in a three dimensional colour space. It is taxing on resources to process the image in its full resolution. When objects are relatively far away, and appear small in the visual field, we would like to take advantage of the higher resolution.

The human eye has a region with maximum acuity in the centre of the macular know as the fovea. Motivated by this physiology the above dilemma

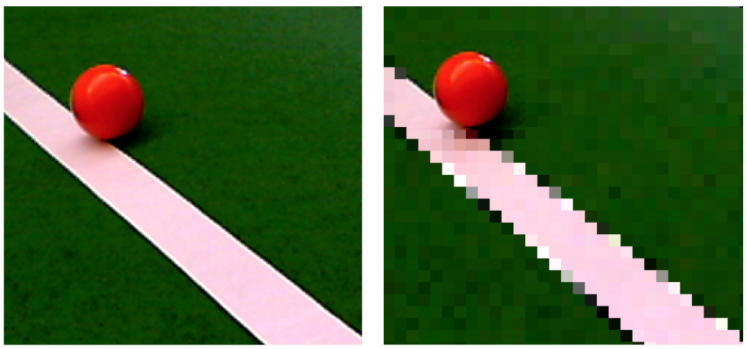

Fig. 1. Foveated imaging. Original Image (left). Virtual fovea centred on a ball (right)

can be addressed by varying the resolution and processing across the image according to one or more points of fixation. This technique is called *foveated imagining*. The fovea provides a high resolution image, but a very narrow field-of-view. *Peripheral vision* is provided by the image outside the foveal regions at lower resolution. These ideas have been used in computer vision inspiring both software and hardware solutions [2]. Figure 1 shows an image at full camera resolution on the left and a foveated image on the right, with the fovea saccaded and fixed on the ball.

The RoboCup soccer environments are characterised by objects with distinct colours. It is not surprising that algorithms to date have largely used colour to identify objects. Organisers have gradually increased the vision challenge by progressively removing crutches such as walls, beacons and coloured goal-posts. In particular, the practice of providing special high luminescent and uniform lighting has been discontinued and robots need to cope with whatever lighting is provided by the venue. Lighting often changes during games as audience numbers fluctuate creating varying overshadowing conditions during the game. One solution is for vision to rely less on colour and more on shape cues.

The contribution of this paper is a vision system that addresses the above needs with the following characterstics:

1. A peripheral vision system to locate salient features. A novelty is the detection of field-edges for localisation using the saliency image alone.
2. Employing foveated imaging techniques to limit resource usage.
3. Relying more on edges and reducing the dependence on colour.
4. Meeting real-time requirements running a close to maximum frame-rate.

The application of these methods have broad applicability. We describe them in the context of the Standard Platform League that uses the small humanoid Nao robot from Aldebaran Robotics. The rules of the league disallow external processing or any modification to the machine. The robots' embedded computer is limited to an AMD Geode LX900 processor running at a modest 500MHz. The playing area of the soccer field is currently 4 by 6 meters with colour coded

open goals. A team-size of three robots was used in 2010 and this will increase to four in 2011. The ball is a standard orange coloured hockey ball. Each robot has two CCD 640×480 pixel cameras in its head (although only one can be used at a time).

In the rest of this paper we will describe the down-sampled "saliency" frames that are used to identify possible locations of various objects on the field. The saliency image is used to find field-edge lines to aid in the localisation of the robot. We next show how the saliency image leads to virtual saccades to multiple points of fixation representing interest regions corresponding to the ball and goals. Multi-modal colour and edge data at high resolution is used at these foveal points in the image, achieving both high accuracy and high efficiency.

This approach was implemented by the UNSW team *rUNSWift* for the Standard Platform League in RoboCup World Competition in Singapore in 2010, for both the technical challenges and the soccer tournament. The University of New South Wales (UNSW) placed first in the technical challenges and second in the soccer competition against 23 other international teams.

2 Saliency Scan

In order to achieve our goal of identifying areas of interest in the image as fast as possible, the first step of the vision pipeline is to subsample the image in a regular gird pattern. We reduce the image size by a factor of n for each of the two image dimensions. We have chosen $n = 4$ for the 2010 competition to make optimally use of machine cycles, but experiments show that $n = 8$ is still acceptable if we wish to free up more resources. The advantage is that the number of pixels to be processes is reduced by a factor of n^2. By mapping every 4th pixel in the raw 640×480 image we derive a 160×120 pixel resolution "saliency image" giving a 16 fold reduction in image size. For $n = 8$ the saliency vision processing load is reduced by 98.4%. Figure 1 (right) shows a down-sampled part-image of the green field, field-line, and ball for $n = 8$. The ball and a small area around the ball shows a virtual fovea region at the original raw resolution.

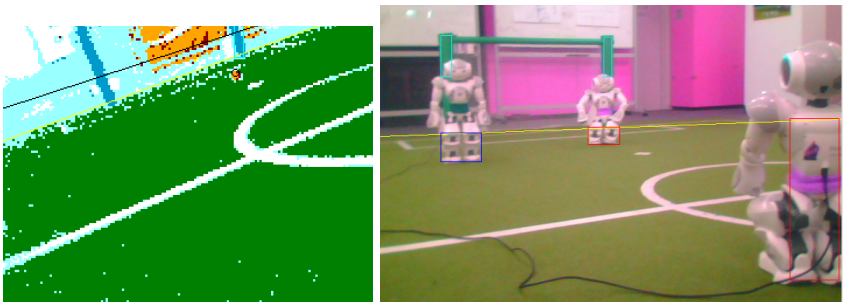

Fig. 2. Colour classified saliency image at 160×120 resolution ie.e $n = 4$ (left). Regions identified during the region detection process (right).

Only a colour classified version of the saliency image is stored. The colours are colour-calibrated off-line using a weighted kernel classification algorithm developed for previous Robocup competitions [6]. This is a nearest neighbour algorithm where each training sample increases a weighting for a particular YUV value toward the classified colour. The classifier is able to generalise to unseen data as neighbouring values in colour-space, within a fixed Euclidean radius, have their weights increased at an exponentially decreasing rate. The kernel file is used to generate a constant-time lookup table on the robot at runtime. The colours calibrated are orange (the ball), green (the field), white (the field-lines and parts of the robots), yellow (the yellow goals), red (the pink band worn by robots on the red team) and blue (the blue goals and the blue band worn by robots on the blue team).

As the saliency image is generated for every visual frame at 30 fps, any further optimisation is desirable. We analysed the compiler-generated assembly code to find other optimisation opportunities. The main optimisation contributions are as follows:

- The histogram data is stored as 16-bit integers, since the saliency image is parameterised by n and can be as large as 640×480 when $n = 1$. It was changed to use 8-bit integers if the image was smaller than 256×256.
- Rather than keeping variables to designate the indices in the saliency image and then iterating through valid values for the indices, a pointer was kept to the active pixels in the saliency image and that pointer was iterated through the entire image.
- Rather than reading each channels of each YUV pixel as three individual 8-bit bytes, the entire 32-bit word containing two VYUY pixels is read, and unused information is removed, thereby requiring one memory access instead of three.
- Our colour classification table is a 2MB $128 \times 128 \times 128$ YUV-to-classified-color lookup matrix. This is now converted to a 16MB $256 \times 256 \times 256$ VYU-to-classified-color lookup matrix. The reason for doing this is so that the conversion from a 32-bit word containing two VYUY pixels to a classified colour can be done in two assembly instructions.
- Rather than storing histogram data for all colours, we only store histogram data for the colours where the histogram is used, i.e. blue and yellow. This requires us to perform a comparison and a conditional branch of every classified pixel, but saves us from performing memory accesses to update the x and y histograms.

While the saliency scan is being built, the body exclusion information is used to remove the robot's own body from the saliency image. The saliency scan is provided an array of coordinates that define the lowest coordinate in each column of the image that is known not to contain the robot's body. Therefore, the saliency scan is filled down a column as normal until this coordinate is reached. All pixels in the saliency scan below this coordinate in the column are marked as the "background" colour. This means that any processing performed

on the saliency scan later in the vision processing pipeline can stop scanning down a column whenever a background coloured pixel is seen.

In the following sections we will describe how the colour calibrated saliency scan can be used to rapidly identify objects of interest in the image. In addition, while the saliency scan is being generated, histograms in the x and y-axes for each of the major field-space colours are generated. The maxima of these histograms can be found efficiently, allowing the rest of the vision system to analyse only the most interesting parts of the image at the native resolution.

3 Field Edge-Detection Using the Saliency Scan

To further reduce the amount of the image that has to be processed for object identification, and to assist with localisation, the edges of the green field are detected using the Saliency Scan image. In 2009 B-Human used a convex hull algorithm to exclude areas above the field-edge [8], which achieves the first goal of reducing the area of the image to be processed. In 2010 rUNSWift used a similar method of vertical scanning to detect points on the edge of the field, but rather than find an arbitrary convex hull, multiple iterations of the RANSAC algorithm [3] are used to find straight lines. When two field-edge lines are detected, the possible positions of the robot are reduced to 4 hypotheses.

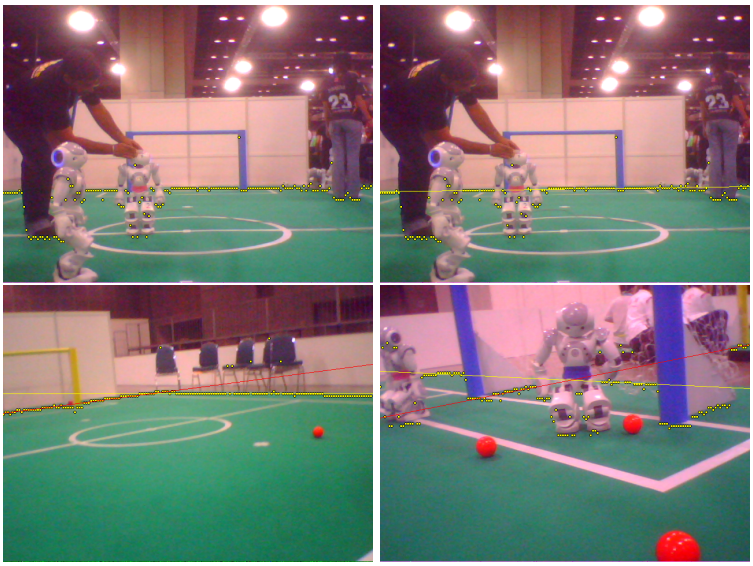

Fig. 3. Candidate points for a field-edge line (top-left). Line found by performing RANSAC on the candidate points (top-right). Lines found by performing RANSAC twice on the candidate points (bottom-left). False-positive field-edge (bottom-right) .

Initially the first green pixel in each column of the saliency scan is recorded, by scanning vertically from the horizon down (the horizon is found by using kinematics to take into account the robot's current joint angles) - Figure 3 (top-left). Secondly, the RANSAC algorithm chooses the parameters for a line in $t_1x + t_2y + t_3 = 0$ form, to maximise the number of points that fit a line - Figure 3 (top-right). Finally, the consensus set of the first line is removed from the candidate points, and RANSAC is repeated, possibly finding a second line - Figure 3 (bottom-left). Figure 3 (bottom-right) shows a false-positive for one of the field-edges caused by the triangluar goal-post support. Its effect is rapidly filtered out with goal-post localisation information.

In addition to reducing the amount of the image to be scanned for objects to the parts of the image below the field-edge, these field-edge observations were able to be used to provide useful updates to the robot's estimated position on the field [1].

4 Finding Interest Points Using the Saliency Scan

We scan the colour classified pixels underneath the field-edge to identify potential areas, or regions, that could represent important features, such as the ball, other robots, or field-lines. The contents of each of these regions are analysed to determine what objects they may represent. By only examining small areas of interest at the full resolution, this method of virtual saccades enabled us to greatly increase the run-time speed of the vision processing system.

Points of interest are found by scanning each column of the saliency scan image below the field-edge to identify runs of non-field green coloured pixels. For runs starting with orange (ball coloured) pixels, the run will finish when either a green, white, robot red or robot blue pixel is found, when a few unclassified pixels are found, or when the bottom of the image is reached. Alternatively, for runs starting with other colours, they will finish when either an orange pixel is found, when more than one green pixel in a row is found, or when the bottom of the image is reached.

Run information is used to build regions. A run is connected to an existing region only if the following conditions are met:

- If the last run added to the region is adjacent to the current run
- If the region contains orange pixels, the run will only be connected if it also contains orange pixels.
- If the run contains robot coloured pixels and the region does not, they are only joined if the region is less than a certain width.
- If the run contains no robot coloured pixels and the region does, they are only joined if the difference between the x coordinate of the current run and the x coordinate of the right most robot coloured pixel in the region is less than a certain threshold.
- If the length of the run is between half the average run length of the region so far and double the average run length of the region so far.

If no region is able to meet all these conditions, a new region is created for the run. An array of pointers to regions containing runs from the previous column is stored to avoid large numbers of regions slowing down processing. The process is summarised in Algorithm 1. An example of the output of the region detection is shown in Figure 2 (right).

Algorithm 1 Region Building Algorithm

```
for all column in saliencyScan do
  for all row in column do
    if have reached the end of a run then
      for reg in lastColumnRegions do
        if reg.startY > run.endY then
          continue
        end if
        if reg.endY ≥ run.startY then
          if conditions for joining run to reg are met then
            if run hasn't been joined to a region yet then
              Join run to reg
              Add reg to end of thisColumnRegions
            else
              Merge reg with previous region run joined
            end if
          end if
        else
          remove reg from lastColumnRegions
        end if
      end for
      if run has not been joined to a region then
        Create new region for run
        Add new region to thisColumnRegions
      end if
    end if
  end for
  Set lastColumnRegions = thisColumnRegions
  Set thisColumnRegions.size = 0
end for
```

Throughout this process, information about each run is collected for the region it joins. Information is collected such as the number of pixels of each colour in the region, the coordinates of the bounding box of the region, the average length of the runs in the regions, the start and end coordinates of each run in the region, and the bounding box of the robot colours in the region to be stored. This information is then used to identify what object (if any) the region is most likely to contain.

The initial object classification is performed by examining the colours, shape and location of each region to determine if the region is more likely to contain

a ball, field line, robot, or just be noise, such as noise from an error in the field-edge. As orange coloured regions are grown separately from other regions, any regions containing orange pixels are considered to be potential balls.

5 Multi-modal Object Analysis in Foveated Regions

An alternative to the use of colour is to use edges to find the outline of objects. Unfortunately, common edge-detection methods to identify all the edges in the image, such as Canney, are computationally too expensive to run in real-time on the Nao, before considering the additional challenge of complex shape identification. A foveated image hybrid solution of these two methods was used to combine the accuracy and robustness of edge-detection with the computational speed of colour calibration to identify both balls and the goal-posts. The hybrid solution involved firstly using colours in the lower resolution peripheral vision to quickly identify salient locations, and then edge-detection to perform accurate and reliable identification at the higher resolution foveated points.

5.1 Ball Detection

For ball detection, edge-detection is used only around the foveated location in the image where a region has been identified as a probable ball. The objective is to find a list of pixels on the edge of the ball. A circle is then fitted to these points to allow the location of the ball to be accurately determined. Rows and columns of the full resolution image are scanned outwards from the region until the v channel of adjacent pixels differs by more than a certain threshold. Only the v channel was used in the ball edge-detection as this chromatic dimension of the ball tends to change quite markedly near the edge of the ball. Edges are often be detected inside the ball when a combination of the y, u and v channels are used.

In order to further increase the efficiency of this method, the space between rows and columns scanned for edges was adjusted according to the size of the region to ensure that balls close to the robot didn't take too long to process, but balls far away from the robot could still be properly identified.

Once pixels around the edge of a ball have been identified, a circle can be quickly fitted to these points by randomly selecting 3 edge pixels, and finding the intersection of the perpendicular bisectors of the lines joining the three points. The intersection gives the centre of the ball, and the distance between the intersection and any of the 3 pixels gives the radius of the ball. If this process is repeated several times and the median of the centre and radius measurements is taken, any small errors in the edge-detection are greatly reduced.

Figure 4 shows an example of the edge-detection being used to accurately identify a ball. The image on the left shows the colour calibrated image. It can be seen that a substantial part of the ball is unclassified (note that unclassified colours appear as light blue in the screenshot). The image on the right shows that the edge-detection has enabled the edge of the ball to be precisely located.

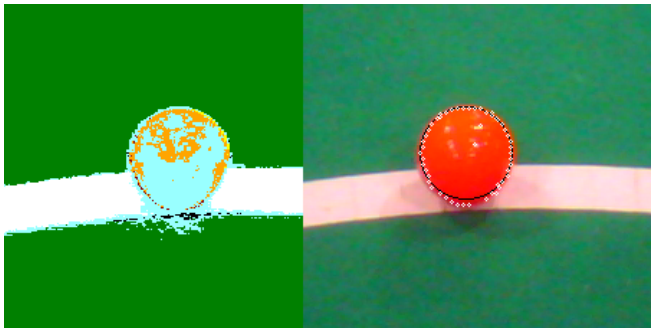

Fig. 4. A screenshot of the ball detection. The left image shows the colour calibrated image, while the right shows the edge points identified and the circle fitted to the edge points.

This is particularly important for ball detection as kicks need to be lined up very precisely for them to work well.

5.2 Goal-Post Detection

As the majority of the goal-posts appear above the field-edge, goals are not identified during region building. Instead, the histograms generated while the saliency scan is being built are used to identify the likely approximate positions of the goals, and edge-detection is then used to find the exact position of the goal-posts, or to remove false positives from the histogram stage.

This is achieved by firstly finding the maximum value in the y-axis histogram for one of the goal colours. Only one y coordinate is used because if there are 2 goal-posts in the image, they will occupy approximately the same y coordinate range, and the maximum in the histogram will most likely occur at a y coordinate occupied by both posts. The x-axis histogram is then scanned to find local maximums above a certain threshold for the goal colour. To avoid several local maximums being detected in the same goal-post, the histogram value of the goal-post colour has to decrease to be at least 3 times less than the maximum value before another local maximum can be recorded. The same procedure is used for both goal colours.

Several horizontal and vertical scan-lines are used around each pair of x and y coordinates identified using the histograms. Each scan starts around the pair of x and y coordinates, and continues outwards until an edge is detected. For goal-detection, an edge is found when the two pixels differ in the sum of the differences in the y, u and v values by more than a certain threshold. All channels are used as the colour of the background around the goal-posts cannot be controlled, so any significant change in any channel needs to be registered as an edge. These scan-lines result in a rectangle representing the goal-post, which can then be used by the localisation algorithms.

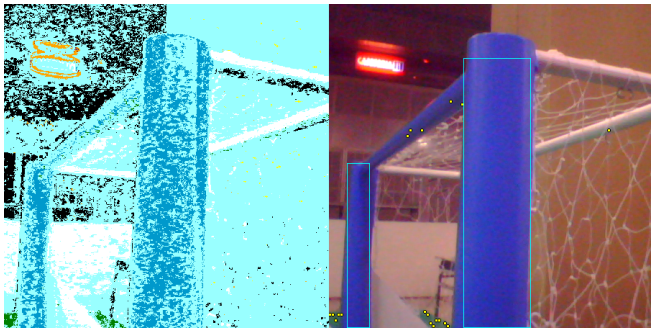

Fig. 5. Poor colour classified image of goal-posts (left). Accurately identified goal-post using edge information (right).

Figure 5 (left) shows a very deteriorated colour classified image of the blue goal. Despite the poor quality, the foveated higher resolution edge-detection approach is able to clearly identify both goal-post, as shown on the right in the Figure.

6 Performance in RoboCup

The set of algorithms presented in this paper form the cornerstone of the UNSW's visual object identification for the 2010 RoboCup competition. In this competition, rUNSWift was placed second in soccer, and first in the technical challenges against 23 other international teams. In particular, the foveated vision algorithm was able to successfully handle the difficult conditions of a final game without noticeable degradation in performance where people crowded around the field creating significant challenges for vision by affecting the lighting. In testing before the competition, we found that vision was able to run at approximately 30 frames per second during game conditions.

As the region builder uses the field edge-detection to only scan the image below the field-edge, and field-edges are used for localisation, field edge-detection is a vital part of our vision system. We found that when the field-edge(s) could be seen clearly, or with a few small obstructions, the field edge-detection worked consistently and accurately. However, when there was a lot of obstruction, such as several robots, or a referee, the field-lines were often mis-placed. At times this caused a noticeable deterioration in the localisation while lining up to make a kick for goals.

The advantage of using the foveated image and virtual saccad approach of initial colour detection, and then accurate edge-detection proved to be very beneficial to the performance of both the gaol detection and the ball detection. In following this method, only a very small number of pixels in the saliency scan needed to be the correct colour for the edge-detection to give an accurate match. This allowed us to consistently and accurately detect the balls and goals, even

from the opposite side of the field despite the large amount of colour variation due to the curved surfaces of the goals and the ball, and various shadows on the goals.

7 Related Work

A number of alternate methods have been devised to solve the complex task of object identification in the resource limited environment of RoboCup.

In order to limit the amount of interference from the background, it is often a useful first step to identify the edge of the field in the image. Any item above this edge can therefore be eliminated. The method used in [8] to find the edge of the field is to scan down each column in the image to find a green segment of a minimum length, and fit a convex hull to the start of the green segments. We imagine this approach would make the position of the field-edge more accurate when there are a lot of objects around the edge, however this would make it much more difficult to use field-edges as part of localisation.

Due to the limited processing power available on the Nao, it is not possible to scan every pixel in the image fast enough to run in real time. An interesting approach is taken in [4], where the density of horizontal and vertical scan-lines is changed depending on how close the scan-lines are in the image to horizon. This uses the theory that objects close to the camera will be large enough to be seen using extremely low resolution scan-lines, but objects further away, near the horizon, will appear much smaller, and therefore need a much higher density of scan-lines in order to be detected. The drawback to this approach is that shape identification and repeated accesses are harder and slower. An alternate approach can be seen in [9], where regions are grown from the green field; with the white field-lines, robots and balls separating the green regions. The authors propose that, as the robot moves, the regions can be incrementally grown and shrunk, resulting in far fewer pixels needing to be processed and updated each frame. This idea of using previous frames to help lower the computation time of the current frame, while not explored in our 2010 vision system, is a worthwhile avenue for future research.

One of the most difficult parts of the object identification for robocup is the distinction between field-lines and robots, as many parts of the robots are white or close to white. This means that some kind of processing, other than colour, has to be used to separate field-lines and robots. The method used in [8] to achieve this is to first create a series of small white coloured regions that could represent either parts of a line or parts of a robot. These regions are then analysed in terms of their shape, and ones that more likely represent robots are marked. Finally, areas of the images where there is a cluster of these marked regions are considered to most likely contain robots, and every region in this area is thus removed. However this method does not actually identify the robots.

The authors of [7] propose a different of edges and colour to achieve fast object recognition. In this method, a grid of horizontal and vertical scan-lines is used to search for pixels where there is a significant drop in the Y-channel

compared to the previous pixels searched. As the field is generally darker than the field-lines and the robots, this can indicate an edge between an object and the field. The pixels around this can then be colour classified to see if they are white or orange.

8 Conclusion

A vision processing system must be highly efficient, robust, and accurate to enable it to perform reliably in the dynamic world of a soccer game. We have presented a foveated imagining approach using colour CCD cameras that can perform the vision task in real-time. We have also presented several processor optimisations to help improve code for low-powered embedded systems. By utilising the hybrid modalities of colour classification and edge-detection, we are able to reliably identify robots, goals, field-lines and balls in the RoboCup environment. Our approach of using virtual saccades to points of fixation of high-resolution foveal areas in the image allowed us to reduce the processing of redundant data, and achieve processing speeds of approximately 30 frames per second in changing lighting conditions.

References

1. Jayen Ashar, David Claridge, Brad Hall, Bernhard Hengst, Hung Nguyen, Maurice Pagnucco, Adrian Ratter, Stuart Robinson, Claude Sammut, Benjamin Vance, Brock White, and Yanjin Zhu. RoboCup standard platform league - rUNSWift 2010. In *Australasian Conference on Robotics and Automation*, 2010.
2. P. Camacho, F. Arrebola, and F. Sandoval. Multiresolution sensors with adaptive structure. In *Industrial Electronics Society, 1998. IECON '98. Proceedings of the 24th Annual Conference of the IEEE*, 1998.
3. Martin A. Fischler and Robert C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24:381–395, June 1981.
4. A. North. *Object recognition from sub-sampled image processing.* Honours thesis, The University of New South Wales, 2005.
5. Board of Trustees. Robocup http://www.robocup.org/.
6. Kim Cuong Pham. *Incremental learning of vision recognition using ripple down rules.* Honours thesis, The University of New South Wales, 2005.
7. T. Röfer and M. Jüngel. Fast and robust edge-based localization in the sony four-legged robot league. *RoboCup 2003: Robot Soccer World Cup VII*, pages 262–273, 2004.
8. Thomas Röfer, Tim Laue, Judith Müller, Oliver Bösche, Armin Burchardt, Erik Damrose, Katharina Gillmann, Colin Graf, Thijs Je ry de Haas, Alexander Härtl, Andrik Rieskamp, André Schreck, Ingo Sieverdingbeck, and Jan-Hendrik Worch. B-human team report and code release 2009. http://www.b-human.de/index.php?s=publications, 2009.
9. F. Von Hundelshausen and R. Rojas. Tracking regions. *RoboCup 2003: Robot Soccer World Cup VII*, pages 250–261, 2004.