

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900

Open access books available

185,000

International authors and editors

200M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Robust and Efficient Robot Vision Through Sampling

Alex North and William Uther

*The University of New South Wales and National ICT Australia
Australia*

1. Introduction

Vision is an extremely important sense for both humans and robots, providing detailed information about the environment. A robust vision system should be able to detect objects reliably and present an accurate representation of the world to higher-level processes, not only under ideal conditions but also under changing lighting intensity and colour balance; when fully or partially shadowed; with specular and other reflections; with uniform or non-uniform backgrounds of varying colour; when blurred or distorted by the object's or agent's motion; in spite of chromatic and geometric camera distortions; when partially occluded; and under many other uncommon and unpredictable conditions. Visual processing must also be extremely efficient, allowing a resource-limited agent to respond quickly to a changing environment. Each camera frame must be processed in a small, usually fixed, amount of time. Algorithmic complexity is therefore constrained, introducing a trade-off between processing time and the quality of information gained.

Within the domain of the RoboCup four-legged league, previous vision systems have relied heavily on the colour of objects since the ideal colour of most important objects is specified in the league rules: a green field with white lines, an orange ball, yellow and blue goals, and pink, blue and yellow navigational beacons. However, there is considerable scope for interpretation and variation allowed by the environmental specification, particularly with regard to lighting intensity and uniformity. Agents must be capable of performing under varying conditions, albeit with time allowed for detailed calibration procedures.

Typical systems group the continuous space of colours returned by the camera into a small set of discrete, symbolic, colours. They then attempt to form objects by grouping neighbouring similarly-classified pixels (Bruce et. al., 2000; TecRams, 2004). In implementation this usually results in a look-up table or decision tree that quickly maps the detected pixel value to a symbolic colour. Typical approaches to the generation of this table involve a supervised machine learning algorithm, where a human expert provides classification examples to a computer program, which generalises these to form the complete segmentation (Pham, 2004; Röfer et. al., 2004; Veloso et. al., 2004; Brusey & Padgham, 1999; Xu, 2004; Chen et. al., 2003).

Source: Robotic Soccer, Book edited by: Pedro Lima, ISBN 978-3-902613-21-9,
pp. 598, December 2007, Itech Education and Publishing, Vienna, Austria

Unfortunately, as lighting conditions change, the colours of real-world objects change and methods relying solely on colour become brittle and unreliable. In addition, these “blob of colour” based methods must process each image frame in its entirety, where there is frequently a large amount of redundant visual information. In contrast, this chapter presents a system that shifts the focus to recognising sparse visual features based on the relationships between neighbouring pixels, and detecting objects from a minimal number of such features.



Fig. 1. A Sony AIBO ERS-7 wearing the red team uniform and the orange ball on the RoboCup four-legged league field. Note the coloured goals and localisation beacons. The outer wall is optional

There have been a number of modifications attempted to address some of these shortcomings with “blob of colour” based vision systems in the RoboCup legged league. A dynamic classification approach, where the classification of a pixel value may change over time, is one such modification. Approaches based on multiple colour segmentations or relative classification have had some success, but such methods rely on multiple fine calibrations or overly simplified representations (Quinlan et. al., 2004; Sridharan & Stone, 2005; Jüngel et. al., 2003; Wasik & Saffiotti, 2002). The conclusion to be drawn from past attempts is that colour segmentation is a difficult problem. Local variations, temporal variation, complexity of segments and overlapping classifications all thwart creation of a perfect static classification, and dynamic classifications have so far been simplistic or drawn unacceptable side effects. The solution lies not in further improving colour classification methods, but in moving away from symbolic colour segmentation towards illumination-invariant vision.

Sub-sampled approaches are a recent development in RoboCup vision. Scan lines over the colour-segmented image are used in (Stone et. al., 2004) for detecting field lines and (Röfer et. al., 2004) and (Velooso et. al., 2004) for object detection. Boundaries are detected between regions of segmented colour lying on the scan lines and classified according to the adjacent symbolic colours. While still dependent on static colour segmentation this method is highly efficient. Along with dynamic colour segmentation, (Jüngel et. al., 2003) recognises that colour relationships are invariant under linear shifts in lighting, and detects edges as *contrast patterns* in the three image channels. Both of these approaches provided inspiration for the procedures presented in this chapter.

This chapter presents an image processing system for the four-legged league based on minimal sampling of the image frame. Rather than process each image in its entirety this approach makes an intelligent estimate of the information content of regions in the image and samples those areas likely to be of importance. Features are detected in the sampled areas of the image and these are combined to form objects at a higher level of processing. Instead of relying on brittle colour segmentations, this approach focuses on the relationships between neighbouring pixels, which remain more constant under variations in lighting.

This approach also presents solutions to, or implicitly avoids, some of the problems identified with purely colour-based approaches: colour segmentation need not be so tightly defined and calibration time is reduced; the calibrated colour relationship tests are environment independent; complex series of manually coded object validity tests are minimised; redundant information is avoided as only information-rich areas are sampled densely; and object recognition is robust to falsely detected features and unexpected background information. This system was successfully utilised by the UNSW/NICTA RoboCup four-legged league team, *rUNSWift*, at RoboCup 2005 and 2006.

2. Theory: Sub-sampled Object Recognition

Our approach is based on feature detection through the use of colour-gradient information gathered with a minimal sampling of each image, and object recognition from these relatively sparse features. It is based on two major theoretical differences from prior systems. Firstly, this approach moves away from absolute colour classification techniques and focuses instead on the relationships between neighbouring pixels. This reduces the dependency of the system on static colour classifications and precisely controlled lighting conditions. Secondly, this approach ceases to process the entirety of each image and instead samples only areas of the image likely to be of high informational value. This aids efficiency and reduces false positive errors caused by unexpected regions of colour.

2.1 Colour Relationships

Static colour segmentations are brittle and depend highly upon the exact lighting conditions under which the segmentation is made. As lighting conditions change the absolute colour values of pixels projected from a certain object change considerably. However, the colour-space differences between pixels representing distinct real-world colours change far less. While pixel values change under varying light, the values for all pixels will change in approximately the same way. Thus, while each colour may no longer be recognisable under a static classification, the difference between two colours remains discernible. Note that the

change in relative values is not exactly linear, depending on the particular characteristics of the light source, so while relative colour is more stable than absolute colour it is not so stable that it remains constant under excessively varying lighting.

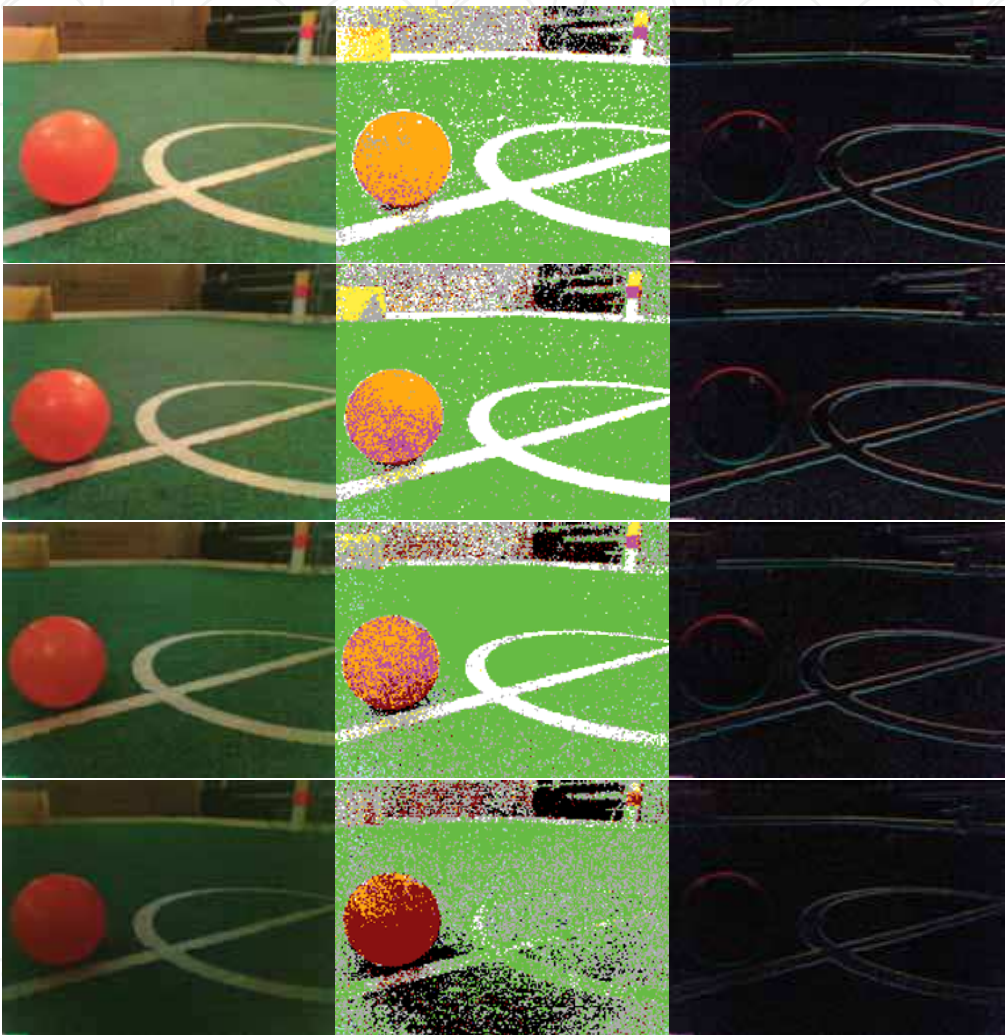


Fig. 2. A scene under progressively darker lighting conditions. The left-hand column shows images from the ERS-7 camera (after correction for chromatic ring distortion). The centre column shows the result of colour segmentation with a colour table optimised for bright lighting. The right-hand column shows the result of applying equation (1) to the images at the left, subtracting from each pixel the value of the pixel immediately above it. Note how this remains stable as the lighting intensity falls, while the colour segmented images deteriorate. The implementation described below successfully detected the ball in all images

$$\begin{aligned}
 Y_{x,y} &= 2 |Y_{x,y} - Y_{x,y-1}| \\
 Cb_{x,y} &= 2(Cb_{x,y} - Cb_{x,y-1}) - 128 \\
 Cr_{x,y} &= 2(Cr_{x,y} - Cr_{x,y-1}) - 128
 \end{aligned} \tag{1}$$

Consider Figure 2; The left-hand column comprises images as detected by the ERS-7 camera, with ambient light intensity decreasing down the page. The centre column displays a static colour classification of the images in the left-hand column under a calibration tuned for bright lighting. Note that as the ambient light intensity is reduced the images become successively darker and the colours recognised by a static classification become increasingly less accurate. The images in the right-hand column show a graphic representation of the colour-space difference between each pixel and its vertically adjacent neighbour. Pixel values are calculated as given in equation (1). This difference precisely picks out the boundaries between the orange ball, white lines and green background. The white wall around the edge of the field is also apparent. The upper and lower boundaries of each object appear as different colours; the lower boundaries represent transitions away from green towards orange or white, and the upper boundaries represent corresponding transitions back into green. Searching for particular boundaries in the original images reduces to searching for these particular “colours” over the difference images. Note that these images remain relatively stable as the lighting intensity falls, in contrast to the rapid deterioration of the colour-segmented images in the centre column.

Detecting differences in neighbouring pixels is very similar to edge detection. General edge detection methods such as Roberts’ and Sobel operators (Roberts, 1965; Sobel, 1970) (see Figure 3) are computationally too expensive to execute on every frame from the ERS-7. Instead, we may calculate a simple one-dimensional gradient. A crucial difference between this method and many standard edge detection algorithms is that the direction of the change in colour-space is explicitly captured, and “edges” can be easily classified by the colour difference they represent. Each difference-value depends on only two pixels, in this case a base pixel and the one immediately above it. Thus, to determine the colour-space transition direction at any particular point requires access to only two pixels. Determining the transition directions over a line of n pixels in any orientation requires access to only $n+1$ pixels, an important result for the sub-sampling approach discussed below.

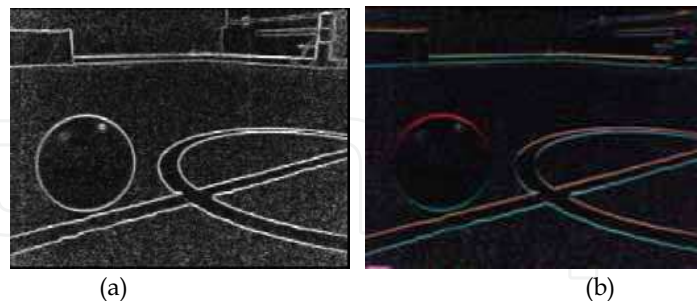


Fig. 3. (a) Roberts’ operator applied to one of the images in Figure 2. The aggregate of the operator result over the three image planes is shown as brightness, normalised to a maximum value of 255. (b) A simple single-pixel difference as in (1) applied to the same image. Both would normally have a threshold applied before use to reduce noise

Since the direction of the vector does not depend on the actual pixel values detected these vectors will be independent of any linear shift of the colour-space. Although no real-world colour-space shift will be perfectly linear, many approximate linearity, being a compression or expansion of the colour-space along one dimension. For example, shadowing or reducing the intensity of ambient light intuitively compresses the set of observed pixel values towards black (i.e., makes them darker, but bright pixels experience greater change than dark ones). This reduction in intensity results in a much smaller change (depending on their magnitude) to the vectors representing object boundaries in the image. Thus, detecting these vectors, rather than specific colour values, is more tolerant of changes in lighting conditions.

2.2 Sub-sampling Images

The information contained in any given image is distributed over its area. This distribution is non-uniform because neighbouring pixels are highly correlated, so the image carries redundant information (Burt & Adelson, 1983). Hence a vision system that processes every pixel in each image necessarily processes redundant information. But which pixels should we process?

In the four-legged league domain, important objects and landmarks are regions of uniform colour. Pixels in the centre of these regions of colour carry little information; their presence can be inferred from the surrounding similarly-coloured pixels. It is the edges of these objects that carry important information about position, size and orientation. Ideally, only pixels near information-bearing edges should be sampled frequently, while pixels in regions of relatively uniform colour should be sampled less often. In other domains the texture of objects may also provide useful information and samples in highly textured regions may provide additional information.

Knowledge of the environment and geometry of the robot allows first estimate for sampling the image. Figure 4 shows the results of applying Roberts' operator to two typical images. Regions of high information are displayed as edges. In general the lower part of an image carries less information (edges) than areas higher up, although the very top part of each frame usually captures background information from outside the playing field.

Objects close to the robot appear lower in the image and larger than objects far from the robot. An object such as the ball carries the same amount of information in either case, but when closer to the robot this information occupies a larger area in the image. Similarly, field lines close to the robot are sparse in the lower areas of the image. Lines further from the robot appear higher up and more closely spaced in the image.

The expected information density reflects this. Figure 5 shows a normalised sum of the results of Roberts' operator over a series of four hundred images captured from an ERS-7, holding its head level while turning its body on the spot at approximately one revolution each 4.5 seconds. The robot is positioned one quarter of the field length from one goal-line and laterally in line with the nearest goal box corner. The result of Roberts' operator on each individual image is thresholded at 40 (of a maximum 255) before being added to the aggregate in order to reduce noise apparent in Figure 4.

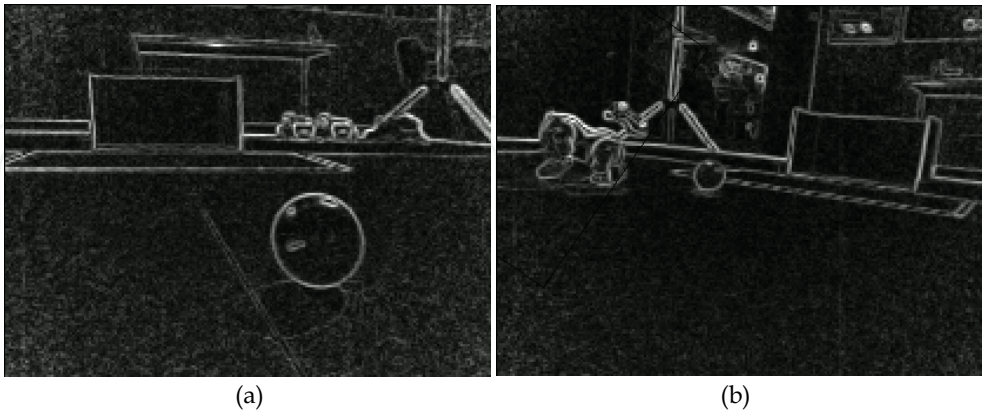


Fig. 4. The results of executing Roberts' operator over a number of typical in-game images. Note that the bottom part of each image, corresponding to objects close to the robot, typically contains sparse information. Objects farther from the robot appear smaller and higher up the image, leading to greater information density near the horizon

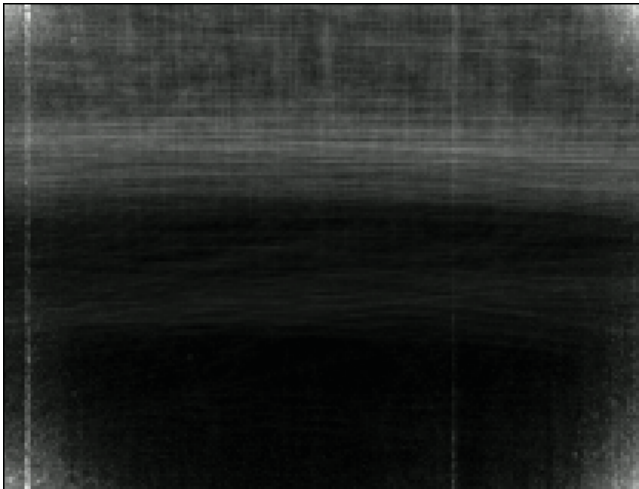


Fig. 5. The aggregate result of Roberts' operator over four hundred images. Note the dark region near the bottom of the image, an area of low average information density. Near the top of the image the brighter region represents an area of higher average information density. Noise due to chromatic distortion correction is visible in the image corners

From Figure 5 it can be seen that the average expected information density is low near the bottom of each image and increases to a peak about one third of the distance from the top of the image. The information density is approximately constant along horizontal lines. This generalisation, that higher parts of an image are likely to carry more information, is only valid when the robot's head is held upright. If the position of the head changes then the

expected information density within the image changes too. This suggests that regions of the image close to the robot's visual horizon should be sampled with higher frequency than areas far from the horizon. We can calculate an artificial horizon from the geometry of the robot's limbs to provide a reference point invariant with its stance.

3. Implementation

We now describe in detail our sub-sampling robot vision system for the RoboCup four-legged league. This implementation is strongly biased towards working accurately, robustly and consistently in the RoboCup competition rather than towards any notion of elegance or mathematical correctness. Despite goals of domain independence implied above, advantage is taken of any reasonable domain-specific assumptions that may be made. This implementation was used with considerable success in the RoboCup 2005 and 2006 competitions.

3.1 Scan Lines

Selection of which pixels to process is by means of a horizon-aligned, variable-resolution grid placed over each image, similar to that proposed in (Röfer et. al., 2004). An artificial horizon is calculated from knowledge of the geometry of the robot's stance and camera. The horizon represents a line through the image with constant elevation equal to that of the camera. This horizon provides a reference point that is invariant with the stance of the robot and aligns the grid with the high-information areas of the image as described in section 2.2.

A scan-line is constructed perpendicular to the centre of the horizon line, continuing down to the lower edge of the image. Scan lines are then constructed parallel to this first line on either side of it at a fixed spacing of sixteen pixels; scan lines continue to be constructed at this fixed spacing until such lines lie entirely outside the image frame. Between each pair of these "full" scan lines a parallel line segment is constructed from the horizon line with a fixed length of 64 pixels. These "half" scan lines are thus eight pixels from each of their neighbouring full scan lines. Between each of these half scan lines and their adjacent full scan lines a "quarter" scan line is constructed with a fixed length of 48 pixels, spaced four pixels from the scan lines on either side. Finally, beginning sixteen pixels below the horizon and doubling this gap for each line, scan lines are constructed parallel to the horizon line until such lines lie entirely below the image frame.

This constructs a grid as shown in Figure 6 (a), with a greater density of scan lines closer to the horizon and more sparsely spaced lines further from it. This grid is used for detection of features related to the ball, field, lines and obstacles. The majority of the scan lines are perpendicular to the horizon, running from pixels that project close to the robot to pixels that project further away. The few scan lines parallel to the horizon are placed to capture features on field lines aligned with the camera axis that would otherwise lie entirely between scan lines. Since no features are expected to appear above the horizon, no scan lines are created above it.

A separate set of scan lines is constructed for detection of landmarks. Beginning just below the horizon line and continuing with an exponentially increasing gap above it, scan lines are constructed parallel to the horizon line as shown in Figure 6 (b). These lines are used for detection of the landmarks and goals around the field.

This same pattern of scan lines may be constructed regardless of the camera orientation, as in Figure 6 (c). In cases where the camera is rotated further than one quarter-turn about its axis the lower edge of the images becomes the upper edge, as seen in Figure 6 (d); in such cases scan lines continue to be processed in the same order and direction. The horizon may not appear within the image in situations where the camera is tilted up or down in the extreme. In these cases an approximation is made, drawing the horizon along the top or bottom edge of the image.

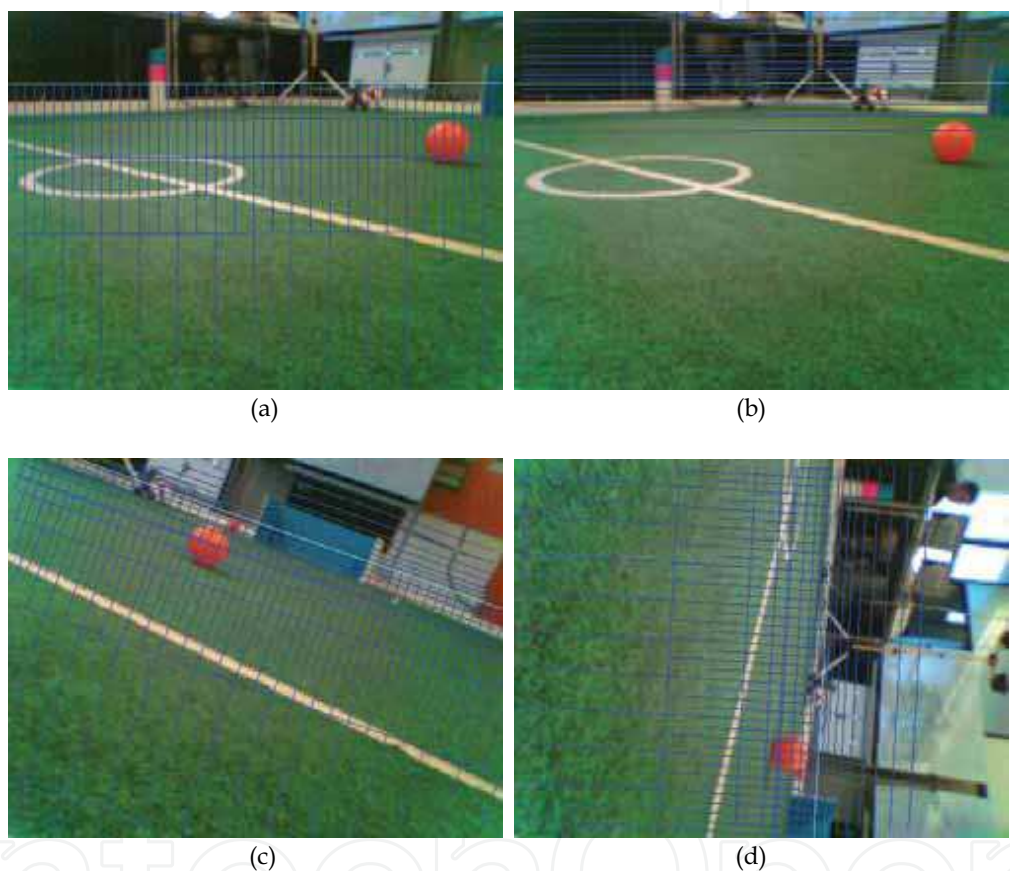


Figure 6. (a) The pattern of scan lines (shown in blue) used to find features on the ball, field, lines and obstacles. The horizon is shown as a pale blue line. (b) The pattern of scan lines used to search for landmarks. (c) The scan lines are constructed relative to the horizon. (d) When the camera rotates more than 90° about its axis the scan lines run up the image

This grid of scan lines defines the initial access pattern for pixels in the image. If no features of interest are detected then these are the only pixels processed in the image. The scan lines typically cover about 6,700 pixels, or twenty percent of an image. More pixels in the vicinity

of features and objects of interest are accessed during the image processing but the majority of pixels are not processed at all.

Some scan lines are also constructed dynamically. After processing of the initial scan line pattern described above, more information is available about the information distribution over an individual image. It is likely that additional processing in the vicinity of the previously detected features will lead to detection of further features, since there is significant spatial locality in edge information. Hence, if fewer than seven ball features (see section 3.3.1) are found in the initial pass then an extra fixed-size grid of scan lines is constructed around each of the detected features.

3.2 Colour Classification

Although this implementation attempts to move away from reliance on statically classified colour such classifications nonetheless provide useful information. Given the specification of the four-legged league domain in terms of objects of uniform colour, symbolic segmentation is used extensively in landmark detection and provides useful confirmation of features detected through other methods. However, only those pixels selected for examination by the sub-sampling approach are ever classified.

Because of the poor quality sensor on the ERS-7, correction is applied for chromatic distortion (Xu, 2004) and the particular robot used to capture images is treated as the reference robot for linear adjustment between robots as described in (Lam, 2004). These corrected images are then loaded into a classification application for classification.

A simple weighted Parzen kernel classifier provides segmentation. Our experience with this classifier suggests that incremental, online training is more useful than increased classification accuracy.

3.3 Edge Feature Detection

Feature detection takes place over the scan lines described in section 3.1. Features of the ball, field, lines and obstacles are detected in the vertical and low horizontal scan lines. These scan lines are processed from bottom to top (or left to right) and each pixel is compared with the previous pixel in the scan line. Rapid changes in the Y, Cb or Cr values indicate edges in the image, and particular edge vectors are recognised as being features of a particular environmental object.

3.3.1 Detecting Ball Features

Ball features represent transitions between the orange ball and the green field or white field lines. Occasionally transitions into the yellow or blue goal are also encountered and the majority are correctly detected by the procedure outlined below.

For each pixel in turn running up the scan line let (y, u, v) be the values of the Y, Cb and Cr channels respectively for this pixel, (dy, du, dv) be the difference between the channels of this pixel and the previous (i.e., next lower) pixel, and sum be the sum of the absolute values $|dy| + |du| + |dv|$. A ball feature satisfies:

1. $sum \geq 20$ (a minimum gradient over all channels), and
2. $|du| > 15$ (a minimum gradient in the Cb channel), and

3. either:
 - a. $dv = 0$, or
 - b. $|du/dv| < 4$ (slope of Cb less than four times slope of Cr), or
 - c. $\text{sign}(du) = -\text{sign}(dv)$ (Cb changes in opposite direction to Cr).

In order to avoid confusion between strong specular reflection and white field lines a ball feature must also satisfy $y < 180$. Note that these rules capture transitions both towards and away from orange without any explicit notion of orangeness.

These features are then confirmed by consultation with the statically classified colours of nearby pixels. While processing a scan line, information about the direction of the scan line in the image is calculated. For these tests this is simplified to the nearest basis direction: up, down, left or right. The value of du calculated above gives the direction of change at this pixel: if $du > 0$ then the transition is towards orange; if $du < 0$ then the transition is away from orange. A line of five pixels beginning at the pixel under test and progressing in the direction of the centre of the ball (simplified to a basis direction) are examined. In order for the transition to be confirmed as a ball edge the classified colour of these pixels must satisfy:

1. at least three are classified orange, and
2. no more than one is classified red, and
3. no more than one is classified pink, and
4. no more than one is classified yellow.

A consistent run of orange is not required; any three from the five pixels may be orange. Thus we confirm that there are at least some orange classified pixels where they would be expected. Note that the pixel under test is not required to be classified orange; in fact it is quite often the case that the very edge pixels are somewhat blurred and take on a classification of white or yellow.

From the images in Figure 2 it can be seen that pixels near the upper edge of the ball maintain the correct classification under the widest range of lighting intensities. This test therefore favours transitions at the upper edge of the ball. The lower part of the ball deteriorates to pink or red quite quickly, and indeed this is a major problem with purely colour-based blobbing approaches. This test will continue to recognise edges so long as three out of five pixels are classified orange, but thereafter will reject the transitions as likely spurious. Heavily shadowed edges on the lower half of the ball are detected with a colour based approach outlined in section 3.4.3.

These colour-based tests are necessary because the transition tests are overly lax. The complexity of the transition tests here can be likened to the simple thresholding that was previously used for colour segmentation. The region satisfying the tests is not well fitted and includes many transitions outside the desired set. This method was chosen because it is simple enough to provide an easy implementation and demonstrate the validity of the concept without introducing many unnecessary complexities; future approaches could adopt more complex transition definitions to reduce reliance on colour segmentation even further.

3.3.2 Detecting Field Lines

Field line features represent transitions between the green field and white field lines or boundaries. For each pixel in turn (after the first) running up the scan line define (y, u, v) , (dy, du, dv) and sum as for ball feature detection in the previous section. A field line feature satisfies:

1. $|y| > 32$ (a minimum value of Y), and
2. $|dy| > 15$ (a minimum gradient in Y), and
3. $|du| < 40$ (a maximum gradient in C_b), and
4. $|dv| < 40$ (a maximum gradient in C_r), and
5. $|du| < 4$ or $\$sign(du) = sign(dv)\$$ (a small gradient in C_b , or C_b and C_r slope in the same direction).

The direction of change may be calculated in a similar fashion to that for ball features, using the indicator dy rather than du . No symbolic colour tests are applied to field line edges. Note that this test is likely to misrecognise the boundary between robots and the green field as being a field line edge; no attempt is made to prevent this since in our implementation such noisy data is handled robustly in the localisation module (Sianty, 2005).

In addition to detecting field lines, a sparse sampling of each image is made for the purpose of detecting the green of the field, as an aid to higher-level localisation systems. Every 32 pixels along half of the full scan lines a check is made: four line segments of nine pixels with a mutual intersection at the centre of each line segment form an axis-aligned star shape containing 33 pixels. If at least two-thirds (22) of these pixels are classified as green then the centre pixel is marked as a *field-green* feature.

3.3.3 Detecting Obstacles

The method for detecting obstacles presented here differs from most other attempts. This approach specifically detects the shadows on the field caused by objects lying on it. While processing a scan line the Y value at each pixel is tested against a threshold value (we used 35). If the Y channel falls below this threshold the pixel is classified as an obstacle. A maximum of five pixels are classified as obstacles on any one scan line.

To avoid false obstacles caused by the ball, the robot itself or human referees a state machine keeps track of the classified colour of pixels on the scan line as they are processed. The twenty pixels immediately above a candidate obstacle are tested: if more than ten green classified pixels or five orange classified pixels are encountered then the obstacle candidate is discarded.

The results of ball, line and obstacle detection are shown in Figure 7.



Fig. 7. The results of feature detection over a sample image. Ball edge features are displayed as yellow points, line edge features as light green points, obstacles as red points and field-green features as dark green points. Black points represent features that have been discarded due to checks applied after initial detection

3.4 Symbolic Colour Feature Detection

As noted earlier, colour remains an important part of the four-legged league domain, and this approach continues to use symbolic colour classification for detection of beacons and goals, using a sub-sampled approach rather than blobbing. Landmark detection takes place over the upper horizontal scan lines described in section 3.1. These scan lines are scanned left to right and each pixel in turn classified into one symbolic colour via a static colour segmentation as outlined in section 3.2. Colour is an appropriate indicator for the landmark objects as they are far less subject to variations in lighting during a four-legged league match than on-field objects such as the ball. Although the colour segmentation must be tuned for a specific environment, the perceived colour of beacons and goals changes little during the course of a match.

3.4.1 Detecting Landmarks

A state machine tracks the number of consecutive pixels found of each of pink, yellow and blue along horizontal scan lines, along with the start and end points of these runs of colour. Up to one pixel of “noise” (any other colour) is tolerated. Beacons are detected by the pink square that appears on all beacons. A run of five consecutive pink pixels (plus one pixel of noise) in a scan line creates a *beacon feature*. Goals are detected by their uniform colour of pale blue or yellow. A run of twelve blue or yellow pixels (plus one pixel of noise) creates a blue or yellow *goal feature*. These features are passed to the object recognition system for further processing.

A slight modification to the thresholds is made when the robot’s head is held low and the horizon coincides with the top of the image, such as when the robot’s head is controlling the ball. Pixels near the top border of the image are subject to significantly more noise (mainly

due to chromatic “ring” distortion and its correction) than those near the centre of the image, and are more likely to be misclassified. Thus, when the robot holds its head down over the ball and is searching for the goal an additional “noise” pixel is allowed in goal features.

3.4.2 Detecting the Wall

While the area outside the green carpeted field is undefined by the four-legged league rules there is often a low wall or region of uniform colour surrounding the field. It is advantageous to detect this to allow filtering of features and objects appearing off-field. A state machine tracks the number of green, grey or white, and other-coloured pixels encountered during the scanning of each vertical scan line. A wall feature is detected by a series of four green classified pixels (allowing one pixel of noise) followed by a series of at least five white or grey classified pixels, followed by a series of two pixels of other colours (allowing two pixels of noise). The requirement for non-green above the white pixels prevents close field lines being detected as walls. A wall feature is created midway between the start and end of the white/grey pixel series. A wall line is constructed using a random sample consensus, or RANSAC, algorithm (Fischler & Bolles, 1981).

3.4.3 Detecting Faint Edges

There are a number of cases where edges in the image become blurred so the ball and field line feature detection methods outlined above become less effective. The most common cause is motion blur: when either the camera or objects in the environment move quickly the result is a blurred image with indistinct edges. In such images the thresholds for change required for feature detection may not be met, since the transition is spread over many pixels. In these cases an alternative ball feature detection method is used, based on segmented colour. While symbolic colour classification is susceptible to changes in lighting it is fairly robust to blurring; edge detection exhibits the opposite tendencies.

A state machine keeps track of the number of orange, maybe-orange (i.e. pink, red and yellow) and non-orange (the remainder) pixels detected along a scan line. A transition is detected between non-orange pixels and orange pixels, possibly with a number of intervening maybe-orange pixels. Three consecutive orange pixels are required to satisfy as an orange region, although the number of maybe-orange pixels before this is unbounded. If the transition is into an orange region a feature is created at a point midway between the last detected non-orange pixel and the first orange or maybe-orange pixel. If the transition is away from orange a feature is created at a point midway between the last detected orange pixel and the first non-orange pixel, so long as these two points are within six pixels of each other. On transitions away from orange the maybe-orange pixels are ignored since such pixels usually occur on the lower half of a valid ball.

3.5 Object Recognition

Object recognition takes place over the features as recognised in the previous section. Object recognition involves grouping features related to the same real-world object and extracting the important attributes of these objects such as position in the image, position in the environment, heading, elevation, orientation and variances over these attributes. The results

of the early stages of object recognition are used to focus computational effort when few features are initially detected.

3.5.1 Beacon Recognition

Beacon recognition comprises grouping the detected pink features into candidate beacons then searching above and below the pink region for the other characteristic beacon colour (pale blue or yellow). Beacon features (which are horizon-parallel line segments) are grouped by merging adjacent “overlapping” features. Two features overlap if a line perpendicular to the horizon can be drawn that intersects with both features.

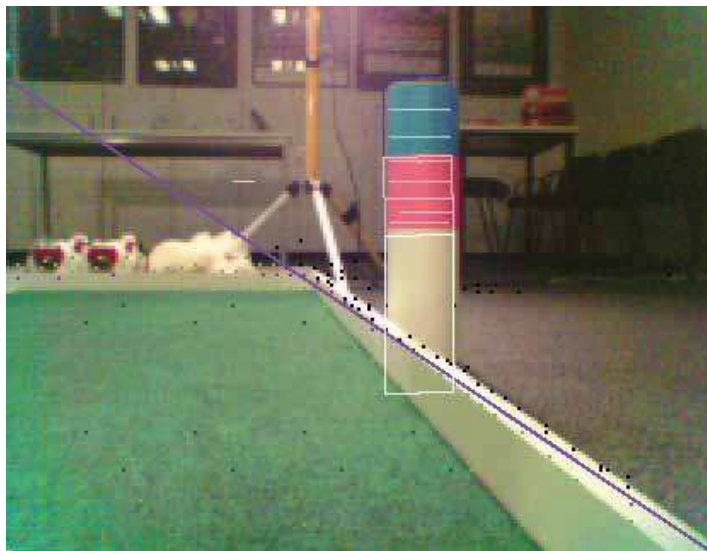


Fig. 8. A recognised beacon. The beacon features are displayed as horizontal pink lines. The white field wall has also been detected, displayed as a purple line

A local search is then performed to classify each group of beacon features as one of the four possible beacons, or as falsely detected features. Essential properties such as apparent height and heading are deduced from the beacon’s geometry through simple methods which are not relevant here. Only a single check is performed to confirm the validity of a candidate beacon (c.f. the sixteen checks listed in (Lam, 2004)). The centroid of the beacon must not be below the horizon by more than 25 pixels. This check rules out some invalid beacon features that might be detected by excessive pink occurring in the ball or red team uniform.

3.5.2 Goal Recognition

Goal recognition comprises grouping the detected blue and yellow features into candidate goals. Goal features are grouped by merging adjacent “overlapping” features in the same way as beacon features, relaxed to allow up to one scan line separating features to be merged. However, it is possible for two distinct regions of the one goal to be visible as

shown in Figure 9. Thus goal feature groups are also merged if they contain features on the same scan line.



Fig. 9. (a) A recognised goal. (b) The goal is often occluded by a robot, but grouping of features leads to correct recognition, even if the goal is divided in half. The two possible gaps for shooting are indicated by horizontal white lines. The recognised goal is assumed to be aligned with the horizon, so no attempt is made to detect the goal outline

As for beacons, the essential properties of a goal are deduced from its apparent geometry through simple methods that are not relevant here. A few checks are made to confirm the validity of a candidate goal: the aspect ratio of the goal is checked to make sure it forms a sensible shape; a goal must not appear further than twenty pixels above the horizon; and a number of pixels underneath the candidate goal are tested for colour. If very few are found to be green the goal is rejected. The goal is also rejected if many are found to be white, as might occur in the blue or yellow patch of a beacon. The aspect ratio and colour checks are ignored when the robot holds its head down low while controlling the ball: both are likely to trigger falsely, and goal detection is of utmost importance in these cases.

3.5.3 Ball recognition

Ball recognition is performed after beacon and goal recognition has completed, allowing obviously spurious ball features to be ignored. If fewer than seven valid ball features have been detected additional scan lines are first created and scanned near existing ball features as outlined in section 3.1.

Ball recognition involves estimating the outline of the ball from the detected features. A circle is fitted to the ball edge features using a generalisation of Siegel's repeated median line fitting algorithm to circles, as described in (Mount & Netanyahu, 2001). Under the assumption that the majority of the points to fit lie on a circle, this algorithm claims robustness to up to 50% outlying data. Slight modifications are made to account for the fact that, due to motion blur, the ball frequently does not appear perfectly circular. This approach assumes that there is at most one ball in view.

Given the parameterised equation of a circle as $(x - a)^2 + (y - b)^2 = r^2$ all triplets of features (i, j, k) are considered, from a minimum of four features. Each triplet determines a circle by

the intersection point of perpendicular bisectors constructed to the chords formed by the triplet. The parameters (a, b) are calculated separately as in (2): for each pair (i, j) take the median of the parameter over all choices for the third point, k ; for each i take the median parameter over all choices for the second point, j ; and take the result as the median over i .

$$a = \text{med}[i] \text{ med}[j \neq i] \text{ med}[k \neq i, j] a_{i,j,k} \quad b = \text{med}[i] \text{ med}[j \neq i] \text{ med}[k \neq i, j] b_{i,j,k} \quad (2)$$

In contrast to the algorithm presented in (Mount & Netanyahu, 2001) the radius is calculated from a single median after the positional parameters have been calculated, as given by (3). This aids stability in the presence of a large number of outliers. If at least seven features are present then the middle three are averaged to give the radius. This averaging helps to reduce jitter induced by image noise.

$$r = \text{med}[i] r_i \quad (3)$$

As for landmark features, the important properties of the ball may be derived from its position and size. Three checks are performed to ensure that the recognised ball is valid. Two geometry tests are applied: the ball is discarded if it appears above the horizon by more than ten pixels; and it is discarded if its radius is unreasonably large (two thousand pixels). Finally, a symbolic colour based test is applied: a valid ball should contain some orange pixels within its circumference. This test is only a validity check; the coloured pixels are not used for deriving the ball's properties.

If the ball is small (a radius of fewer than ten pixels) then a square of side length equal to the ball's radius is constructed around the centre of the ball, and all pixels in this square are colour-classified. Otherwise, features that lie within ten pixels of the circumference are randomly chosen and a line segment is constructed between the feature and the ball centroid. The pixels along this line segment are classified, up to a total of one hundred pixels over all features. In both cases counts are maintained of the number of orange, red, pink and green classified pixels encountered. If fewer than three orange pixels are encountered the ball is discarded. If the radius of the ball is greater than ten pixels and fewer than twelve orange pixels are found the ball is discarded. This colour checking is displayed as a cross shape of accessed pixels over the ball in Figure 11.

This approach allows accurate recognition of the ball under a range of conditions. While it is limited by an assumption that there is only one ball present in the image, the ball may be detected when blurred or skewed, occluded or only partially in frame. Figure 10 shows recognition in a number of these cases. The repeated median algorithm exhibits $\Theta(n^3)$ computational complexity in the number of features. Since n is usually small this remains appropriate; this implementation limits the number of features used to a maximum of seventeen, more than enough to achieve an accurate fit.

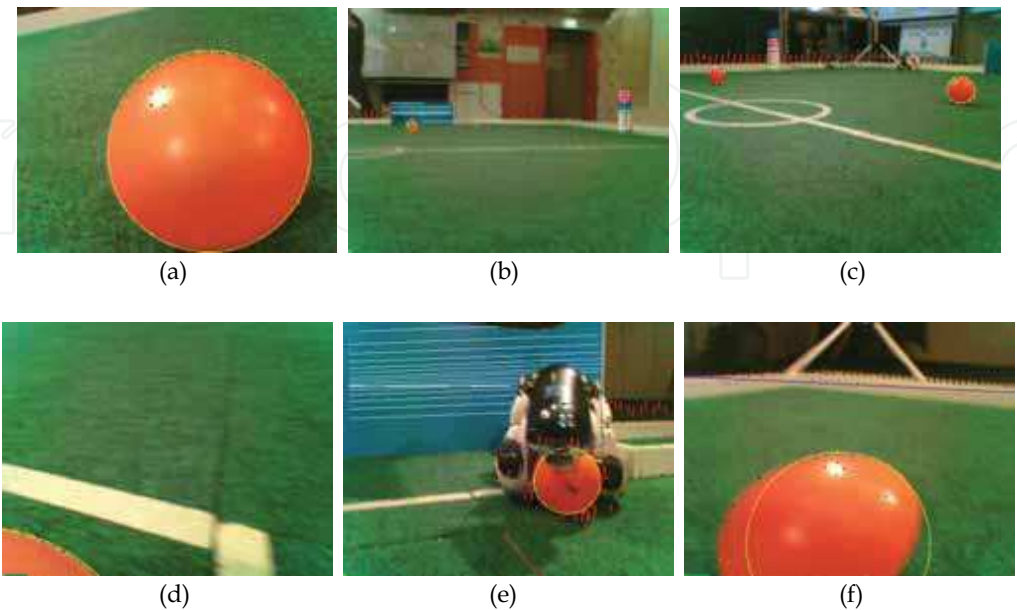


Fig. 10. Ball recognition in a number of different situations. Ample information available in (a) allows for a very tight fit. The repeated median estimator continues to be accurate at long range in (b), although at such distances precise information is less important. Noise features caused by a second ball in (c) are ignored. A ball lying mainly outside the image frame in (d) is still detected accurately, as is a ball partially occluded by a robot in (e). A combination of motion blur and skewing in (f) lead to a questionable fit

3.5.4 Mutual Consistency

Once object recognition is complete a small number of checks are made to ensure that the perceived objects are mutually consistent. These checks are outlined below, in order of application.

1. The two goals cannot be simultaneously perceived. If they are, the one comprising the fewest features is discarded. If they have the same number of features the goal with the highest elevation is discarded.
2. A goal centroid cannot appear inside a beacon. If it does, the goal is discarded.
3. A beacon centroid cannot appear inside a goal. If it does, the beacon is discarded.
4. A goal cannot appear above a beacon by more than ten degrees. If it does, the goal is discarded.
5. Diagonally opposite beacons cannot simultaneously be observed. If they are, both are discarded.
6. Beacons at the same end of the field cannot appear within thirty degrees of each other. If they are, both are discarded.
7. The ball cannot appear above a goal. If it is, the ball is discarded.

8. The ball cannot appear above a beacon. If it is, the ball is discarded.

These checks conclude visual processing for one frame. The information extracted from the recognised features and objects is passed to the localisation and behaviour modules.

4. Evaluation

In our domain of robot soccer, the accuracy and robustness of a vision system reflects strongly in the performance of a team of robots in a competitive soccer match. In a limited sense this is the most valuable method of evaluation. The goal of the vision system is to have robots play the best soccer and a vision system that results in a team consistently winning matches is better, in some sense, than a system that does not. However, it is difficult to hold other variables constant, so, while being the most important test of a system's quality, this test is also the most subject to random variation, noise and external influences. The performance of a team depends upon the performance of the opposing team and the environment both on and off the field. Further, this method of evaluation is highly non-repeatable; it is impossible to substitute an alternative vision system and have the same match play out with the exception of changes directly related to vision. Nevertheless, evaluation by playing matches remains an important measure of progress. If otherwise identical code is used in both teams over a number of competitive matches the influence of the vision systems may be observable in qualitative terms.

Behavioural evaluation of a single robot agent is another important method of evaluation. Agent provides information about the performance of its vision system in particular circumstances, with much of the interference caused by team-mates and opponents removed. For example, an agent's behaviour may provide clear indication of whether or not it can see a given object. It is possible to display informative indicators in the form of LEDs, or such information might be accessible via a remote data stream. Thus the quality of a single agent's visual information may be subjectively assessed. Alternatively, two independent agents might be active simultaneously, and the behaviour and data streams from each compared. Although each agent is processing different input, over time any significant systematic differences in visual processing will become apparent.

Single agent tests bear some semblance of repeatability: situations can be constructed and the performance of agents evaluated over a number of similar trials. Single agent tests are particularly useful for evaluating small modifications to a vision system. The RoboCup four-legged league also presents a number of technical challenges that are useful in evaluating an agent's vision system (RoboCup, 2005).

One of these was the *Variable Lighting Challenge* which explicitly tests an agent's vision system's robustness to changes in lighting conditions over time. In this challenge an agent must consistently recognise the standard RoboCup objects while the field lighting changes in different ways. While still heavily dependent on higher level behaviours this challenge tests robustness of image processing systems to shifts in lighting intensity, colour temperature and dynamic shadows. For an agent to perform well in this challenge it necessarily requires a highly robust vision system.

For yet more detail, an agent's image processing system may be compiled and executed in isolation on a standard PC ("offline"), where its performance for particular images or image sets may be qualitatively and quantitatively evaluated. This allows direct observation of the

performance of a system in precise circumstances, and in many cases provides insight as to why a system behaves as it does. A log file of the image data as sensed by the camera may be captured then played back, allowing the performance and internals of the system to be visualised and examined in detail.

Unlike the two previous methods, offline evaluation is fully repeatable and allows multiple image processing systems to process exactly the same data and the results to be compared with each other and a human-judged ground truth. This allows direct comparison of similar vision systems, and comparison against a subjective ideal interpretation. Alongside single agent evaluation this method is effective at comparing modifications to a vision system.

These three evaluation methods present trade-offs between their importance, accuracy and ease of administration. This approach, like many others, is designed to perform as well as possible in one general environment, here the competitive environment of the RoboCup four-legged league. In one sense this real-world performance is the most important evaluation measure.

The *rUNSWift* team using this system placed first in the RoboCup 2005 Australian Open and third in the RoboCup 2005 four-legged league World Championships. In 2006, *rUNSWift* were again Australian Champions and came second in the international RoboCup competition. The extremely successful GermanTeam also use elements of sub-sampling and colour relationships (Röfer et. al., 2004), so these methods are clearly a valid and successful approach to four-legged league vision. While it is extremely difficult to directly compare the accuracy and robustness of different vision systems, this chapter presents some results pertinent to this approach.

4.1 Colour Relationships

A focus on the relationships between neighbouring pixels, rather than symbolically classified colour, leads to a great improvement in robustness. As demonstrated in Figure 2, variations in lighting quickly lead to the degradation of static colour classifications, while colour-space relationships remain far more constant. This was confirmed by the ability of the system presented here to perceive the ball and field lines in a range of environments without re-calibration. In fact, the colour-difference vectors in this implementation were not modified after their initial calculation despite large changes in lighting intensity, colour temperature and even camera settings across four different field environments in which the implementation was tested.

This system was also used without significant modification for the 2005 four-legged league Variable Lighting Challenge, in which *rUNSWift* placed third in the world. The *rUNSWift* agent was able to perceive the ball for the majority of the challenge, especially as light intensity fell. However, when lighting was made significantly *brighter* than normal game conditions some red/orange confusion hampered performance. These performances demonstrate robustness to dynamically variable lighting.

The reduced reliance on colour segmentation has also led to reduced complexity of and effort required for colour segmentation. Coupled with an interactive classification tool and the efficiency and flexibility of the kernel classifier, colour segmentation can be performed with more sample data and with less effort than previous approaches. The sub-sampling system is robust to a sloppy segmentation, although in a competitive environment effort should be made to achieve a stable classification. The final rounds of the RoboCup 2005

four-legged league competition were performed with a segmentation based on over one thousand images, trained in a few hours of human expert time and tested only in the few hours before the competition games.

4.2 Sub-sampling

Focussing image access on regions of high informational value leads to efficiency gains, as time is not spent processing redundant information within uniformly coloured regions. Domain knowledge allows targeting of high information areas for dense sampling, while regions of typically low information may be sampled more sparsely. A dynamic element to image sampling allows even more directed processing based on the information gained in earlier stages. Most potential invalid object candidates are implicitly rejected: only regions of an image likely to contain a particular feature are sampled for it, reducing the number of invalid objects it is possible to recognise falsely.

Pixel access to a typical image is shown in Figure 11, where it can be seen that areas containing useful information are sampled with higher density than less informative regions. The scan line pattern typically covers only twenty percent of the pixels in an image, with dynamic access processing a little more depending on the information gained in the initial pass.

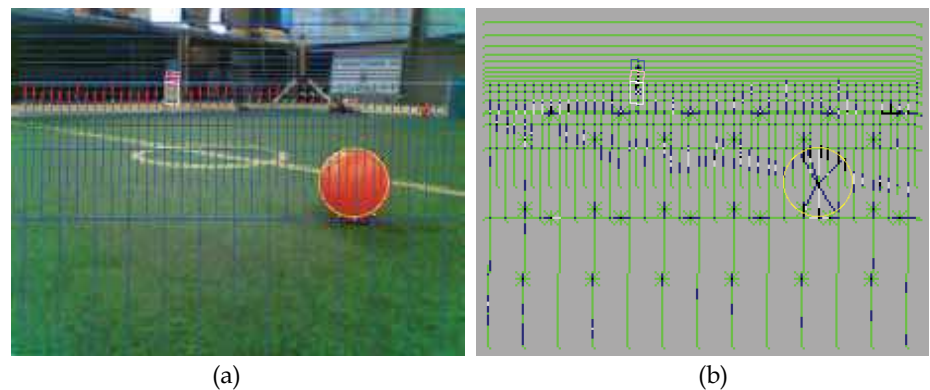


Fig. 11. Pixel access profiling for a typical image. (a) shows the original YCbCr image with scan lines, features and recognised objects displayed. A pixel access profile is shown in (b). Dark grey represents pixels that have not been accessed, green represents pixels that have been accessed once, and other colours represent pixels accessed multiple times. Note how access is clustered around high-information areas of the image such as the ball, field lines and beacon

Actual processor time consumed by this approach is in fact similar to previous methods, with complete visual processing typically ranging from 10ms – 15ms, depending on image content. There are a number of reasons why this is not significantly lower. Firstly, the scan line pattern describes essentially random access to the image. Approaches that sample the entire image may make good use of processor caching and similar optimisations, but these are lost when the image data is accessed in this pattern. Similarly, image correction and colour classification are applied on the fly to pixels as they are accessed, giving random

access to correction and classification tables. Disabling chromatic distortion correction leads to a small gain in efficiency. Removing all colour classification-based access leads to large gains but hampers the still colour-based beacon and goal detection. Finally, it is advantageous to use all available compute power to construct the best possible information, so optimisation efforts were not made beyond those necessary to reach this level of performance.

4.3 Object Recognition

Moving to a sub-sampled approach makes traditional blobbing approaches to object recognition impossible. Instead, more sophisticated object recognition procedures are required to form objects from sparsely detected features. A significant advantage to this approach is the reduction in complex, hand-coded validity tests for objects. (Lam, 2004) explicitly listed thirty such checks involved in blob-based detection of the beacons, goals and ball for *rUNSWift* in 2004, but the code used at the 2004 RoboCup competition contained many, many more and spanned thousands of lines of code. The fifteen validity tests outlined in section 3.5 represent the entirety of such checks in this implementation; in general this approach requires far fewer domain-specific tests. This leads to more efficient object recognition, allowing for more sophisticated and computationally expensive approaches to obtaining accurate information.

4.4 Shortcomings

As noted above, while the efficiency of the sub-sampling system as implemented is well within acceptable ranges for the four-legged league, improvements may be made by optimisations to the image access pattern or a reduction in colour-based tests. Random access to both the image data and correction and classification tables make poor use of processor caching features.

The beacon detection implemented in this approach leaves room for improvement. Being colour-based, it still has the deficiencies associated with reliance on a finely tuned static colour segmentation, but makes use of less information than other approaches. While the accuracy was adequate for our purposes there are gains to be made in more accurately fitting the beacon model to the available data. However, it is likely that the coloured beacons will be removed from the four-legged league field definition in the near future as the league attempts to move towards yet more realistic environments.

Edge-based methods in general respond poorly to blurred images, which are not uncommon for legged robots. Significantly blurred images, such as those obtained while contesting positions with robots from the opposing team, are poorly processed. Section 5.1 suggests some possible improvements.

5. Conclusion

The high level of detail and dependence upon environmental conditions makes creation of an accurate, robust and efficient robot vision system a complex task. Rather than a traditional focus on colour segmentation of entire images, the vision system outlined in this chapter moves towards detection of local relationships between a subset of the image pixels.

While systems based on colour segmentation are brittle and respond poorly to variations in lighting, the relationships between colours exhibit independence to lighting conditions, leading to a more robust system.

This chapter has demonstrated the stability of colour relationships under variations in lighting, particularly intensity. This system continues to provide reliable information under a range of environments and variations in lighting conditions, leading to a more robust feature detection system. A reduced dependence on colour segmentation also leads to reduced calibration time and eases the transition between different environments.

The effectiveness of sub-sampled image processing approaches has also been demonstrated. The information contained in a typical image is not uniformly distributed over its area; neighbouring pixels are highly correlated. In order to more effectively make use of constrained resources, regions of the image typically high in information content are sampled more densely than areas of typically low informational value. A dynamic element to the sampling allows an even tighter focus on useful regions in any given image. The sub-sampled approach leads to efficiency gains and implicit rejection of much unwanted data.

Given the information provided by a sub-sampled, edge-oriented approach, this chapter has also described robust recognition of objects for the four-legged league from relatively sparse features. Object recognition is performed over a discrete set of features corresponding to particular features in the sampled areas of each image. Domain knowledge allows accurate recognition under a range of conditions.

The successful results obtained by the approach presented in this report outline a path to more robust, lighting-independent robot vision. While there is still much work to be done, significant improvements in robustness have been gained by a shift of focus away from statically classified colour towards detection of colour relationships and transitions. Unlike approaches based on selection between multiple colour tables this approach gracefully caters for unexpected conditions without the need for additional calibration efforts. In contrast to existing dynamic classification approaches, this implementation allows for potentially arbitrary complexity in colour and colour-gradient classification without the need for adjustments calculated from past observations.

These changes in focus are likely to be applicable to other robotic vision domains where uniform colour is a primary differentiator for important objects. It is immediately applicable to other RoboCup leagues, and to other domains requiring robust object recognition under tight constraints on efficiency.

5.1 Future Work

While the system as implemented has been successful, a number of areas may provide fruitful future research. The image access patterns described in this report focus on areas of typically high information, but the concept can be taken further. A focus on dynamic processing, where image access is determined by information obtained by previous processing, could lead to even further gains in efficiency. The scan lines themselves could be sub-sampled, performing some variation of an interval search for transitions, sampling every pixel only around areas containing edges. In addition, temporal awareness might be used to further hone access patterns; areas of recently high information value might be sampled first and more densely than areas of little recent value.

Blurring remains a problem for transition-sensitive techniques. Consulting the relationships between pixels at greater intervals than immediate neighbours might allow detection of softer edges. A blurred transition between two colours will have a similar profile to a sharp transition if viewed over more widely spaced pixels. Conversely, on sharp transitions more detail is available from the ERS-7 camera than is currently used. The Y channel is sampled at twice the resolution of the chroma channels, and this information might be used to improve the accuracy of the location of detected features.

Perhaps the most clearly beneficial direction for future work is in generalisation of the colour gradient space regions used for feature classification. Strong parallels may be drawn between colour segmentation and the classification of transitions, the major difference being that the transitions are invariant under linear shifts in the colour-space. The classification procedure for colour-gradient vectors used in this implementation is of a similar complexity to early colour segmentation routines. Applying present-day colour segmentation methods to gradient classification would likely lead to a great improvement in the accuracy of detected features and further reduce reliance on colour segmentation.

Finally, there are some possibilities for improvement upon the object recognition methods presented in this report. The assumption that there is only one ball, and that it appears as a circle, limits both the flexibility and robustness of ball recognition. Some variation on feature clustering would serve the dual purposes of allowing recognition of multiple balls and rejection of gross outliers. The repeated median circle estimator is effective but its computational complexity prevents use of abundant features. The addition of an optimisation step such as least squares approximation may prove more efficient and allow fitting of more general ellipsoids.

National ICT Australia (NICTA) is funded by the Australian Government's Department of Communications, Information Technology, and the Arts (DICTA) and the Australian Research Council through Backing Australia's Ability and the ICT Research Center of Excellence programs.

6. References

- Bruce, J.; Balch, T & Veloso, M. (2000). Fast and inexpensive color image segmentation for interactive robots, *Proceedings of the 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS-00)*, volume 3, pp. 2061–2066, 0-7803-6348-5, Takamatsu, Japan, October 2000, IEEE Computer Society, Los Alamitos, CA, USA
- Brusey, J. & Padgham, L. (1999). Techniques for obtaining robust, real-time, colour-based vision for robotics, *RoboCup-99: Robot Soccer World Cup III*, Veloso et. al. (Ed), 63-73, Springer, 3-540-41043-0, Berlin
- Burt, P. J. & Adelson, E. H. (1983). The Laplacian pyramid as a compact image code. *IEEE Transactions on Communications*, Vol. 31, No. 4 (April 1983), pp. 532–540, 0096-2244
- Chen, J.; Chung, E.; Edwards, R.; Wong, N.; Hengst, B.; Sammut, C. & Uther, W. (2003). *Rise of the AIBOs III - AIBO revolutions*. Technical report, 2003, The University of New South Wales, Sydney, Australia
- Fischler, M. A. & Bolles, R. C. (1981). Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, Vol. 24, No. 6 (June 1981), pp. 381–395, 0001-0782

- Jüngel, M.; Hoffmann, J. & Löttsch, M. (2003). A real-time auto-adjusting vision system for robotic soccer, *7th International Workshop on RoboCup*, Polani et. al. (Ed.), 214-225, Springer-Verlag, 978-3-540-22443-3, Berlin
- Lam, D. (2004). Visual object recognition using Sony four-legged robots. Honours thesis, 2004, The University of New South Wales, Sydney, Australia
- Mount, D. M. & Netanyahu, N. S. (2001). Efficient randomized algorithms for robust estimation of circular arcs and aligned ellipses. *Computational Geometry*, Vol. 19, No. 1 (June 2001), pp. 1-33, 0925-7721
- Pham, K. (2004). Incremental Learning of Vision Recognition using Ripple-Down Rules. Honours Thesis, 2004, The University of New South Wales, Sydney, Australia
- Quinlan, M. J.; Murch, C. L.; Moore, T. G.; Middleton, R. H.; Li, L. A. Y.; King, R. & Chalup, S. K. (2004). The 2004 NUbots team report. Technical report, 2004, The University of Newcastle, Newcastle, Australia
- Roberts, L. G. (1965). Machine perception of three-dimensional solids. *Optical and Electro-Optical Information Processing*, J.T. Tippet et. al. (Ed.), pp. 159-197, M.I.T. Press, Cambridge, Massachusetts
- Röfer, T.; Laue, T.; Burkhard, H.; Hoffmann, J.; Jüngel, M.; Göhring, D.; Löttsch, M.; Döffert, U.; Spranger, M.; Altmeyer, B.; Goetzke, V.; Stryk, O.; Brunn, R.; Dassler, M.; Kunz, M.; Risler, M.; Stelzer, M.; Thomas, D.; Uhrig, S.; Schweigelshohn, U.; Dahm, I.; Hebbel, M.; Nistico, W.; Schumann, C. & Wachter, M. (2004). GermanTeam 2004. Technical report, 2004, Universität Bremen, Humboldt-Universität zu Berlin, Technische Universität Darmstadt, University of Dortmund
- Shammy, J. (2005). Real-Time Shared Obstacle Probability Grid Mapping and Avoidance for Mobile Swarms. Honours thesis, 2005, The University of New South Wales, Sydney, Australia
- Sianty, A. (2005). rUNSWift RoboCup Challenges 2005, Honours thesis, 2005, The University of New South Wales, Sydney, Australia
- Sobel, I. E. (1970). Camera Models and Machine Perception. PhD Thesis, 1970, Stanford University, California, USA
- Sridharan, M. & Stone, P. (2005). Towards illumination invariance in the legged league. *RoboCup 2004: Robot Soccer World Cup VIII*, Nardi et. al. (Ed.), pp. 196-208, Springer, 978-3-540-25046-3, Berlin
- Stone, P.; Dresner, K.; Fiedelman, P.; Jong, N. K.; Kohl, N.; Kuhlmann, G.; Sridharan, M. & Stronger, D. (2004). The UT Austin Villa 2004 RoboCup four-legged team: Coming of age. Technical report, 2004, The University of Texas, Austin, USA
- Veloso, M.; Chernova, S.; Vail, D.; Lenser, S.; McMillen, C.; Bruce, J.; Tamburrino, F.; Fasola, J.; Carson, M. & Trevor, A. (2004). CMPack'04: Robust modelling through vision and learning. Technical report, 2004, Carnegie Mellon University, Pittsburgh, USA
- Wasik, Z. & Saffiotti, A. (2002). Robust color segmentation for the RoboCup domain, *Proceedings of the 16th International Conference on Pattern Recognition (ICPR-02)*, pp. 651-654, 0-7695-1695-X, Quebec, Canada, August 2002, IEEE Computer Society, Los Alamitos, CA, USA
- Xu, J. (2004). rUNSWift thesis - low level vision. Honours thesis, 2004, The University of New South Wales, Sydney, Australia

RoboCup (2005), Technical challenges for the RoboCup 2005 legged league competition.
<http://www.tzi.de/4legged/pub/Website/Downloads/Challenges2005.pdf>

TecRams (2004), Team Report (2004). Technical Report, 2004, Tecnologico de Monterrey
Campus Estado de Mexico, Mexico



Robotic Soccer

Edited by Pedro Lima

ISBN 978-3-902613-21-9

Hard cover, 598 pages

Publisher I-Tech Education and Publishing

Published online 01, December, 2007

Published in print edition December, 2007

Many papers in the book concern advanced research on (multi-)robot subsystems, naturally motivated by the challenges posed by robot soccer, but certainly applicable to other domains: reasoning, multi-criteria decision-making, behavior and team coordination, cooperative perception, localization, mobility systems (namely omnidirectional wheeled motion, as well as quadruped and biped locomotion, all strongly developed within RoboCup), and even a couple of papers on a topic apparently solved before Soccer Robotics - color segmentation - but for which several new algorithms were introduced since the mid-nineties by researchers on the field, to solve dynamic illumination and fast color segmentation problems, among others. This book is certainly a small sample of the research activity on Soccer Robotics going on around the globe as you read it, but it surely covers a good deal of what has been done in the field recently, and as such it works as a valuable source for researchers interested in the involved subjects, whether they are currently "soccer roboticists" or not.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Alex North and William Uther (2007). Robust and Efficient Robot Vision Through Sampling, Robotic Soccer, Pedro Lima (Ed.), ISBN: 978-3-902613-21-9, InTech, Available from:
http://www.intechopen.com/books/robotic_soccer/robust_and_efficient_robot_vision_through_sampling

INTECH
open science | open minds

InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2007 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](https://creativecommons.org/licenses/by-nc-sa/3.0/), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.

IntechOpen

IntechOpen