2004 UNSW Robocup Team
Thesis A Report

Kevin Chan, Daniel Lam, Derrick Whaite, Ted Wong, Jing Xu

Contents

1 Introduction

This report is an account of what has been done by the 2004 RoboCup Team, the results of The Australian Open 2004 and plans of work of rUNSWift team in the coming 2 months before the world final RoboCup in Lisbon, Portugal. The RoboCup Team 2004 was formed in late November 2003, and most of the work concern porting the robot codes from the old robot model ERS-210 to the new model ERS-7, which include re-writing modules of actuator control, low and high level vision, and behaviour.

1.1 Vision

One of the robot's main sensors is the camera in its nose. The images taken by the camera are processed by the vision module and information about the outside world are derived accordingly. Each image's YUV components are looked up in a precompiled colour table and is associated with a symbolic colour such as field green or red robot. Then this information is grouped by the blobber[1] and verified by sanity checks[2]. The details of the colour segmentation and object recognition process can be found in Chapter 2 of the 2003 thesis[1].

The images taken by the camera are highly vulnerable to noise; also, since the rUNSWift vision system is heavily dependent on colour, it is highly sensitive to shadows and changes in lighting. Standard image processing techniques were applied in an attempt to alleviate the above mentioned problems. A good overview of many of these techniques can be found in the ImageJ documentation[3].

A significant problem introduced by the Sony AIBO ERS-7 camera are chromatic aberrations near the edge of images (approx 30% of the total area depending on colour), in the form of a "blue ring". The concept of electric field lines is used to model this effect, background information regarding electric fields[4], and an implementation of a Java demo program[5] were essential to the development of this solution.

Although the ERS-210 model also suffered from chromatic aberrations, the extent was much less severe. A comparison of chromatic aberrations between an ERS-210 robot and an ERS-7 robot can be found on Roefer's website[6].

1 rUNSWift's colour segmentation module
2 rUNSWift's decision process for recognising colour blobs as real objects

1.2 Localisation

The rUNSWIFT 2003 report [1] discusses the previous system used by the UNSW team, which is the base upon which the 2004 team have built. The localisation system, explained in chapter 3, uses Gaussian probability distributions, an extended kalman filter used for robot position tracking, and the information form of the kalman filter used for opponent and ball tracking. All localisation were based on the beacons and goals. While effective with last year's rules and AIBO model, localisation performance degraded somewhat this year due to beacon removal, poor colour resolution and extra noise present in the new AIBO model's camera.

Having backup position estimations can provide some security against wrong observations. Jensfelt and Kristensen [7] present and demonstrate global localisation of a robot using a multiple hypothesis kalman filter, ie. the robot's state probability distribution is represented by a set of weighted Gaussian distributions. In the algorithm they present, each hypothesis is copied before their kalman correction, if the probability of the update matching the hypothesis is above a threshold value. If it is below the threshold, no update is done on the old hypothesis, but the new update is kept as a new hypothesis. While implementing a similar system would add robustness, it wouldn't utilise any extra localisation information.

In the new rules for the Robocup 2004 competition, the two mid-field beacons have been removed, and the white walls that surround the field have been approximately halved in height. As a consequence, the robots on field have less information and a greater chance of interference from background objects when determining their field position and heading. To help supply more localisation information, field line / border information can be used. In 2003, the German Team demonstrated line detection localisation using a particle filter, or "Monte Carlo Localisation", described in [8]. Accuracy was sufficient to play a full Robocup game without using beacons, although using a lot of processing power. The filter worked by assigning each particle a probability which determines how well it matched the lines seen. The particle distribution was then renormalized so that each particle had equal weight. Some reasons for this method's effectiveness were: it does not rely on a single starting point for edge localisation, and can approximate any form of probability distribution. With its potential demonstrated, an attempt to implement a field line and border localisation algorithm was made.

1.3 Behaviour

The rUNSWift behaviours are specified in a large decision tree. The decision is constantly evolving as there are modifications made each day in order to correct or improve the behaviour in a certain game situation. However, all versions of the trees share common features such as, hierarchically determining the robot's states, roles (eg. defensive, supportive, offensive), the main strategies of specific roles, and different skills and actions to be applied. The nodes of the decision trees consider a number of factors, for example, the positions of all robots, the distances and heading of different objects, and the probabilities of their locations. The desired joint values and wireless information are then communicated to the other team mates via the wireless network.

The current strategies and behaviours are largely based on that of the last two years, and their details can be found in the last two years' team reports [1][2].

2 Work Since 2003

2.1 Vision – Low Level

2.1.1 Colour calibration

Given the importance of the colour look up table in the object recognition process, much time was spent on calibrating it to be as accurate and robust as possible. The process starts with taking a set of images of objects of interest (Table 2.1 of the 2003 Thesis[1]) as training data. Then interesting objects in each image were hand labelled with their corresponding symbolic colours, this information is in turn processed by a machine learning algorithm (C4.5) which outputs the relationship between a colour's YUV values and its symbolic colour as a decision tree. Finally, a look up table is generated accordingly.

The ratio of different labelled symbolic colours in the training data can have a direct impact on the accuracy and robustness of the robot's vision, and such ratios can only be learnt through experience.

2.1.2 Image processing

Many image processing techniques were applied in an attempt to reduce an image's (a) noise and (b) sensitivity to changes in lighting. Techniques such as gray morphology, smooth, sharpen and blur were used in an attempt to reduce noise, but the benefits did not justify the high performance cost of convolution-based implementations.

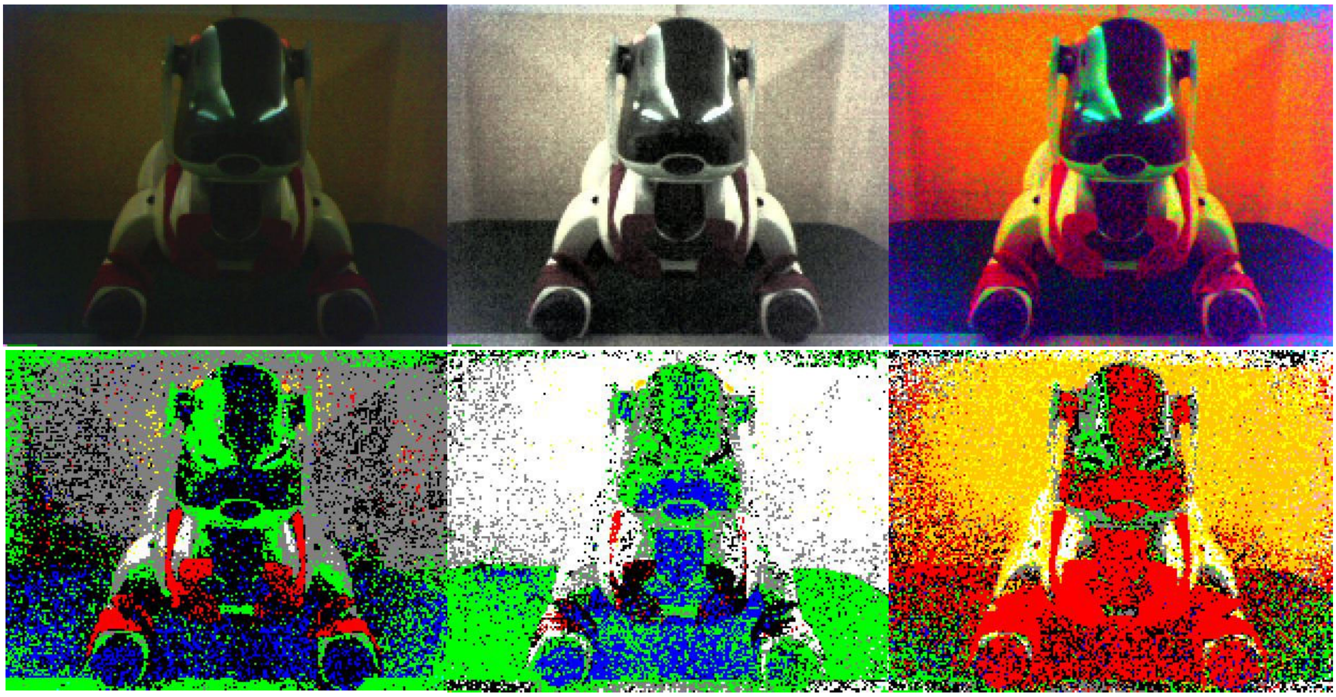


Figure 1 Histogram Equalisation
(a)Original dark image of red robot in yellow goal (b) Equalised Y plane only (c) Equalised all YUV planes
(d)C plane of original image (e) C plane of equalised Y plane only (f) C plane of equalised all YUV planes

One way to combat problems associated with lighting changes, is to adjust the contrast of each image. Histogram equalisation was tested, and reliable contrast could only be generated for the Y plane. When histogram equalisation was only applied to the Y plane of a dark image (Figures 1a & 1d), objects in that image would be more recognisable to the human eye only (Figures 1b & 1e), they still lacked enough colour (UV) information to be recognised correctly by a robot. However, when all three YUV planes were equalised (Figures 1c & 1f), there was too much colour for a robot to recognise objects correctly.

An implementation of histogram equalisation which assumed temporal locality was created, it was fast enough to be deemed feasible to use in a real-time robot.

2.1.3 Ring correction

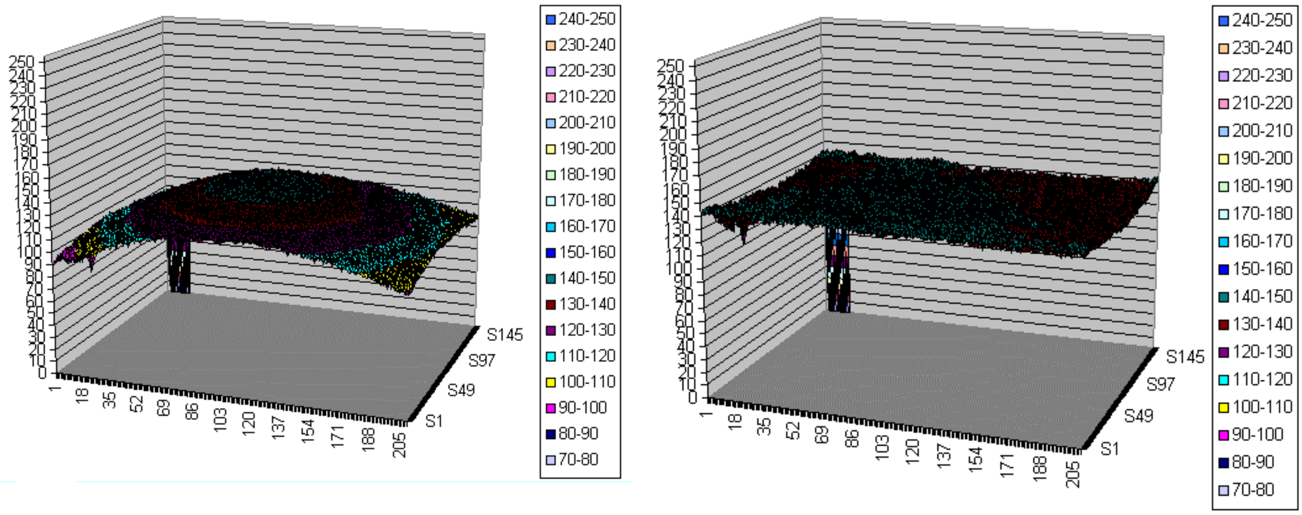


Figure 2 Uniform colour – white border (a) Unprocessed Y plane (b) Processed Y plane

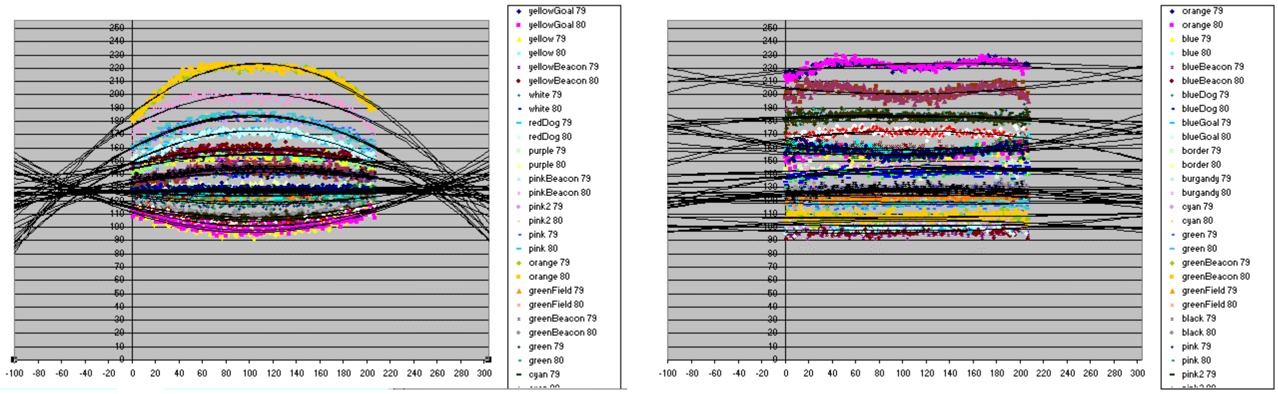


Figure 3 Cross section of uniform colours at Y=79 and Y=80 (a) Unprocessed U plane (b) Processed U plane

The chromatic aberrations near the edge of images darkens the edges of images to an extent such that objects affected by the ring are seen in the wrong colour and are hence recognised as either something else (false positives) or not recognised at all (true negatives). This created problems such as phantom blue robots in the four corners of an image, frame-dropping occasionally from the extra noise, and not being able to utilise the goalie's ability to see two beacons at the same time. After careful analysis of images of uniform colours (Figure 2a) and their cross sections (Figure 3a), it was found that the distortions could be roughly modelled by electric field lines generated from two equal but opposite point charges. However, the distortions were more curved than electric field lines, so a higher gradient was used. After correction, although some of the uniform colours (Figures 2b & 3b) still retained some distortions due to "kinks" in the original image, they are no longer significant. A more comprehensive comparison can be found in Appendix A.

2.2 Colour Segmentation

In the 2003 report[1] the students stated that the blob algorithm was buggy because often the orange ball blob on the CPlane[3] was incorrect. However this is a misunderstanding, the blob algorithm was correct but some minor rounding off errors on the CPlane made the blob display on the CPlane look incorrect. In 2004 a new blob algorithm was written. This algorithm produces exactly the same output as the 2003 algorithm, however it is slightly faster, especially when there are lots of frame noise. The new algorithm uses the Disjoint Sets Algorithm, where each segment is treated as a disjoint set. Each disjoint set has one root, where the root represents the whole set. The set merging process is very efficient. The advantage of this algorithm is that there is no need to examine all segments from the previous rows when merging some segments – only the roots need to be examined, there is absolutely no drawback for this algorithm.

2.3 Vision – High Level

Rather than saying what have been improved since 2003 in the beginning, it might be more descriptive and give a better idea of what has happened to say what has been broken since. There are a few factors that imply sanity checks need to be more robust. First of all, the images from the camera of the new model robot ERS-7 has more edge distortions in comparison to those from ERS-210. This results in (a) Blue robot being recognised in corners of; and (b) Pink blobs of beacons are seen as orange or red when they are in the corners of the frame, and hence recognised as the ball or red robots. Aside from camera distortions, the other factors include the removal of studio lights in the original 2004 rule, which made it difficult to distinguish between black and hence the difficulty in classifying blue robots. Also, the new model ERS-7 has a shiny white coat whose reflections of red uniform and balls look orange, and hence become classified as balls.

In order to compensate for these disadvantages, a few additional checks were made. These include a horizon check by drawing a line in the CPlane and ignoring any object recognised above it. This line is calculated mathematically using the head angles (pan, tilt, crane) of the dog.

[3] Representation of an image in symbolic colours

2.4 Localisation

2.4.1 Gaussian Sum Probability Distribution

The previous localisation system, explained in [1], consists of applying a kalman prediction and correction cycle to a Gaussian probability distribution with mean $\underline{\mu}$ and associated covariance matrix $\underline{\underline{C}}$, where $\underline{\mu}$ is 3-dimensional (2 position coordinates and 1 heading coordinate). The Gaussian sum extension uses n Gaussians, each with a separate mean $\underline{\mu}_i$ and covariance $\underline{\underline{C}}_i$, but also a weight w_i, such that the probability of the robot being at position $\underline{x}$ is:

$$P(\underline{x}) = \sum_i^n \frac{w_i}{\sqrt{2\pi}^3 \sqrt{\det \underline{\underline{C}}_i}} \exp\left(\frac{1}{2} (\underline{x} - \underline{\mu}_i)^T \underline{\underline{C}}_i^{-1} (\underline{x} - \underline{\mu}_i) \right) \qquad \textbf{Equation 1}$$

Predictions and corrections are applied to each Gaussian independently, with the weights w_i staying constant during the prediction (motion update) stage, while being scaled according to observation probability in the correction (observation update) stage. The weight scaling function used is similar to the one defined in the "likelihood" equation (15) of [7]; it is:

$$S_i = \frac{1}{\sqrt{\det \underline{\underline{E}}_i}} \exp\left(\frac{1}{2} (h(\underline{\mu}_i) - \underline{z})^T \underline{\underline{E}}_i^{-1} (h(\underline{\mu}_i) - \underline{z}) \right) \qquad \textbf{Equation 2}$$

where $\underline{\underline{E}}_i = \underline{\underline{R}} + \underline{\underline{H}}^T \underline{\underline{C}}_i \underline{\underline{H}}$ **Equation 3**

When an observation measurement of $\underline{z}$ (with covariance $\underline{\underline{R}}$) is applied to each Gaussian, the weight for that Gaussian w_i is multiplied by S_i defined in (1.2). In the equation, the function $h(\underline{\mu_i})$ transforms the Gaussian's mean to observation space, and $\underline{\underline{H}}$ is that function's Jacobian evaluated at $\underline{\mu_i}$. Following the scaling of weight values, the probability distribution is renormalised by making sure the sum of all w_i is 1. If any weight drops below a threshold value, then the corresponding Gaussian is removed entirely from the distribution, while any two Gaussians with similar means and covariances will be combined.

With the new distribution functions, it is possible to copy existing Gaussians before applying an observation update, and after a time the incorrect Gaussian will be eliminated, which allows some shielding against bad observations (although not against consistently bad observations).

2.4.2 Field Line / Border Localisation

The other attempt to improve the accuracy of robot localisation was a field line / border detection algorithm that utilised a gradient ascent of a match function to estimate the true position of the dog.

Line and border detection begins by scanning the c-plane along a grid of horizontal and vertical lines. While traversing a line, a white – green – white transition marks a field line, while a white – green or green – white transition marks the field border. The position of these transition pixels on the CPlane is recorded. These line points are then projected from the robot's camera plane onto the ground plane, ie. into robot local ground coordinates. Using the robot's current best estimate of its position, these local ground coordinates are transformed to global field coordinates.

How well these points line up with known line and borders on the field is calculated by applying a match function to each point, and summing over all points. There are two match functions: one for field lines, one for borders. They are essentially large 2-dimensional look-up tables that increase in value when the input position is close to a field line or border. When the match value for each point is calculated, so is the match gradient, which is also summed over all points to return a robot position gradient, which is the direction the robot position estimation should be moved in order to increase the match value quickest. The gradient ascent algorithm multiplies this gradient by a constant (the "learning rate"), and moves the robot position by that product. Note that the robot position mentioned includes two spatial and one heading coordinate. It then repeats the process a set number of times.

The final position is used to update the kalman filter. If the global coordinates of the line points are centred around a known line on the field, then a number of possible final positions are generated by code that exploits the known symmetries of the field. This set of final positions is sent to the kalman filter, which can now handle multiple observations due to the new multiple-Gaussian probability distribution code.

It is hoped that the final position is a local maximum of the match function, however this is not always the case. If the gradient is sufficiently small, each iteration of the ascent will only result in small movement of the position estimation, and so the final position may not move far enough to reach the maximum. Another problem is that the heading gradient can vary significantly against the two spatial gradients, and so choosing a fixed learning rate can lead to unstable or insignificant changing of the robot's estimated heading. A combination of these problems resulted in the line detection being too inaccurate for use in the Australian Open.

2.5 Behaviour

Due to various difficulties encountered during the early part of our development, many of the basic skills and motions that were relied heavily on before are now performing in an ineffective manner, which has a huge impact to our strategy. A huge amount of effort has been spent on making these skills workable once again. In fact, all the skills have been partly reworked and a few replaced, or even completely discarded.

There exists several problems that made previous skills and strategies unsuitable in the new robots:

1. The robot weighs heavier than the old model, while we are still investigating a better walk type. So we had frequent hardware failures.
2. The robots parts are larger, hence most skills that involve close contact with the ball, such as grabbing the ball, became ineffective and unsuccessful.
3. Colour distortion, which led to many other side effects, including indecisiveness and incorrect recognition.
4. The removal of the middle beacons due to this year's rule change, so that the decision making process cannot rely too heavily on the localisation unit.

Hence, our effort was oriented towards alleviating these problems and improving the skills, in the hope that we could at least have a sustainable game that could last for a ten minute interval.

First of all, kicks which consume a lot of energy, like the forward and lightning kicks, are avoided because they require too much power when the robot gets up on the ground after finishing the move. Also, they require a very accurate ball-grab (one that must succeed at least 97% of the time) as a pre-condition. On the night before the Australian Open, more paw kick cases were put in to filter out the chances of any forward and lightning kicks.

Ball-grab itself is another problem, since the offset between the robot's front paws and its chest is too short, it leaves no buffer space for the ball to be held. The use of the close range ball sensor does not help the situation either. Yet the head, given now there is an extra degree of freedom and a longer head and neck, is used in grabbing and holding the ball. The chance of a successful grab is significantly increased, though it is still not good enough for any fierce and fast approaches.

A side kick has also been adopted from the University of Pennsylvania team in a complete rewrite of the important turn kick series, which was purposefully crafted for the old robot. Another very important technique, spin-dribbling, was again partly rewritten for the new robot, which would behave well if the ball is correctly captured.

A dynamic motor gain control was also written, such that if the instantaneous battery current is high, all the leg motors are switched to a smaller set of PID gains to avoid sudden hardware failure and overheating, and if it is under a certain stable level, the gains are set to high so that the leg movements are swift.

However, the improvements so far has not been a sufficient solution, as we have observed in the practice matches and in the Australian Open, because they do not bring the essential factors for victory, which are speed and aggressiveness.

2.6 Walk

There were many walking types in 2003, but basically only two walks were used in the 2003 competition – offset walk and canter walk. Offset walk was used for its fast speed and canter walk was used for its stability.

However none of the walk from 2003 can be used in 2004 because of the physical differences in the new ERS7 model. Better and faster walk types were invented. In 2004, three walks were developed – normal walk, high gain normal walk and elliptical walk[9]. Normal walk is a slower walk type which moves approximately 15 cm per second, it is not useful when walking forward because it is too slow, this walk corresponds directly to the 2003 canter walk. Normal walk with high gain is not really a different walk type, it is just the normal walk with high gain. The high gain version produced a much faster walk than the low gain version, however this walk consumed much more battery power than the low gain version and caused the robot to crash more easily. Elliptical walk is a completely new walk, its locus shape is a semi-circle. Elliptical walk is much faster than the normal walk. However the elliptical walk was only successfully developed two days before the Australian open and hence was not used in the Australian Open. Both low and high gain normal walks were tried in the Australian Open.

3 Australian Open

3.1 Introduction

The Robocup Australian Open has been held three times. It is an annual event where the Australian universities compete against one another. However, it is not exclusively Australian, foreign teams have occasionally participated to boost numbers since relatively few Australian universities take part in the Robocup program. The rUNSWift team had won the Australian Open for the first two years, and it was hoped that the tradition could be continued. The Australian Open would also be a good opportunity to test the new ERS-7 robots outside of UNSW's own robot lab, against credible opposition, and with the variations in light and surroundings that a different location brings.

3.2 Vision

The main vision problems in the Australian Open were due to shadows, not enough training data of an unfamiliar environment, and a slightly "warm" colour temperature in lighting. From tests run between matches, it is speculated that the reason robots would back off the ball instead of attacking it was due to seeing phantom red robots in shadowed orange balls and phantom blue robots in shadows on the field, which had the same effect of seeing team mates with the ball. Unfortunately, our machines could not be setup during the matches to log the CPlanes, so we were unable to see how big the red or blue blobs were in the ball. This would have been helpful since the back-off was not triggered as often under the UNSW lab conditions.

Also, UTS had halogen lights which shone directly onto the Robocup field, this added a pink complexion to images, which made it even more difficult to distinguish between orange balls, pink beacons and red robots. In addition, not enough training pictures were taken to generate a more robust colour calibration.

3.3 Localisation

Robot localisation during the Australian Open only used the fixed landmarks, ie. the beacons and goals. Despite this, it worked well enough that the robots rarely got lost. The main reason for this was that the perimeter walls around the field were still the height specified in the 2003 rules, eliminating the false or mistaken beacons that can occur with the walls at the shorter 2004 rules height.

3.4 Behaviour

In the Australian Open, the robots moved too slowly to be competitive opponents. Although a faster and more stable elliptical walk was developed a week before the competition, the strategies that being used were not compatible. Furthermore, we were still unable to entirely avoid the sudden hardware failure in the competition so we were left to use the only choice at that time, which was the improved version of the normal walk.

As observed from the performance of the other teams, such as the University of Newcastle and the Microsoft Hellhounds, they did exploit the potential of the new robot's physical advantage in their robots movements and kicking motions. To combat their faster walk types, the current strategy and skills must be redesigned completely for a more intelligent play.

3.5 Walk

In the Australian Open, UNSW was the second slowest team. Since the elliptical walk was only invented two days before the competition, there was not enough time to integrate this walk in the behaviour and hence it was not used. The slow walking speed was one of the major causes of failure in the Australian Open. In the UNSW lab, the robots sometimes crashed even with the most stable low gain normal walk. With high gain, the robots could not survive for even two minutes. So in the first three rounds of the Australian Open, the robots used the low gain normal walk to minimise crashing. However none of the robot crashed in the first three rounds of Australian Open, so the team decided to switch to the high gain normal walk against UTS.

The results were positive – robots did not crash for the entire game even with high gain. Apparently the UTS lab had a much softer carpet surface than the UNSW lab, with a softer surface the robot had less resistance when walking. We speculate that less walking resistance (or friction) led to less battery current consumed which made the robot less likely to crash. The new OPEN-R released two days before the Australian Open, may have also made the robots less likely to crash.

3.6 Wireless

The TCP protocol was used in the Australian Open but this led to a very serious wireless problem. The robots could not connect to each other with the latest Sony TCP remote processing. Fortunately UTS allowed the UNSW to run remote processing with an older version of Open-R. With the older Open-R, the robots were able to connect to each other. The Newcastle team explained this problem to the UNSW team. The problem caused by changes in remote processing in the latest Open-R, the latest Open-R does not allow the robot to broadcast packets to other robots. One solution is to disable packet broadcasting, so instead of robot A broadcasting a packet to robot B, robot C and robot D, Robot A would send the packet only to robot B, and then robot B forward the packet to robot C, and finally robot C forward the packet to robot D. Although this method would work it is not recommended, since it is possible that Sony will change the remote processing method again, then future teams must re-implement the wireless protocol. A better solution is to switch from TCP to UDP, so that the robots don’t need to use the Sony TCP-gateway.

3.7 Teamwork

The 2004 team has worked cooperatively since its establishment, but some major problems surfaced during the Australian Open. The team lacked organisation when preparing robots for games, causing the first half of the first game to be almost forfeited. This was partly due to a lack of experience in playing matches, and also partly due to a lack of communication between team members. Incompatibilities between different parts of the code were also apparent, eg. the low level vision saw too much red in orange balls, high level vision did not sanity check it out, which caused behaviour to back off balls. This again, was due to a lack of communication and experience.

4 Future Improvements

4.1 Vision – Low Level

Shadows continue to be a major problem in vision, one way to improve this situation is to brighten the images by adjusting the camera settings. The only setting which can be adjusted to make an image brighter is to slow the shutter speed (ie. to increase the length of exposure to light), but currently it is set to fast in fear of distortions caused by motion blur. Motion blur is only significant when a robot pans its head too fast, and this can be detected by the robot. A feasible solution is to adjust the camera settings dynamically depending on how fast the robot's head moves, although the extent of its effect still requires further investigation.

Maybe colours is a concept created by the 2003 team to distinguish between solid colours and colours that could correspond to more than one symbolic colour [1]. It was not successfully implemented last year, but with the use of a new machine learning algorithm LMT, it could now be feasible to implement maybe colours.

The rUNSWift vision system has always been almost completely dependent on colour, but with the poor quality of the ERS-7 camera and the likelihood of changing lighting conditions, it could be feasible to change to a different method of object recognition. A method suggested by the 2002 team was a contrast-based vision system[2], but it has been difficult to have reliable contrast in unprocessed images, and implementations attempted in 2003 using edge detection had proven to be too expensive and inaccurate [1][10]. Improvements in 2004 has made it possible to generate reliable contrast in the Y plane through histogram equalisation. With a combination of simple contrast-based edge detection and colour, it may be possible to develop a contrast/colour hybrid vision system that is more accurate and robust than the current colour-only system.

4.2 Vision – High Level

A few areas have been identified as top priority for improvement, these includes fine tuning the dogs' back-off behaviour and more robust check for beacons. Horizons can to be more accurate because the current calculation has not taken the motion shakes into account. We can take joint angles into account and produce a more accurate horizon calculation.

Also, the classification of dog right now is using a very old method created in 2001 without much revision afterwards. It tries to scan red and blue blobs from left to right, and group blobs base on the horizontal and vertical distance. We can improve the recognition by considering not only from left to right, but also from right to left.

4.3 Localisation

Although localisation performance during the Australian Open was adequate, a workable line detection and localisation algorithm is needed, since it provides advantages such as better goalie positioning, more effective forwards (since they wouldn't have to active localise as often), and effective localisation when beacons are not present or visible (such as in localisation challenges, or if rules eventually change to eliminate all beacons). Ideally, the robot dogs should be able to localise solely of field lines and borders, with beacons / landmarks only used to remove ambiguity between identical lines (due to field symmetry).

The German Team have demonstrated this ability using a particle filter, explained in [8], and so a sensible plan would be to investigate how to incorporate these methods with the current line detection code. One possibility is a hybrid Gaussian-particle scheme, where a number of points, generated from each Gaussian's mean and variance, are tested to see how well they match the edge points seen. Other function maximisation methods, which may or may not use gradient information, should also be investigated.

4.4 Behaviour

One proposal is to use directional and fire paw kicks and the new side kicks and incorporating only the elliptical walk type to achieve speed. The other benefit that we now have is the use of Python script to write the behaviour module, hence saving us time and effort in almost every aspects of development.

Moreover, as wireless network delay is small in the new model, it may be feasible to utilise the wireless communication for team role assignment and determination, instead of just visual information.

Under the non-uniform lighting of the UTS laboratory, our robots recognised the orange ball as red robots and the forward player was backing off most of the time from the false team mates. The back-off and support strategy may need to be reconsidered once again if it persists in the future practice matches. In addition, the robot tended to get confused about its own position in the longer sides of the field. More active localisation skill that also considers the growth of the variance might be needed in the forward player to improve this situation.

4.5 Teamwork

A solution to increase the team's experience with matches is to play regular formal games that mimic game conditions in competitions. A solution to the communication problem is to hold regular meetings where the first part of the meeting is dedicated to important issues, and the second part to open discussion. This should improve both discipline and organisation, and aid in the exchange of ideas between team members.

5 Conclusion

In this report, the work done by rUNSWift team 2004 before the Australian Open in April 2004, the results of The Australian Open 2004, and areas of work that the team considered as top priority were described. Overall, the space for improvement after The Australian Open 2004 is substantial, and whether we can succeed in implementing our plans, and have enough time to test them, will be critical to our performance in RoboCup in Lisbon in July 2004.

References

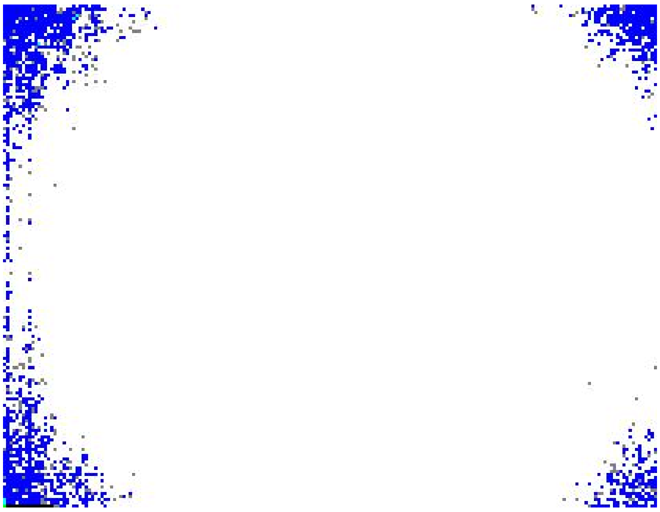
1. Chen, J., Chung, E., Edwards, R., Wong, N., Hengst, B ,.Sammut, C., Uther, W.,
   Rise of the AIBOs III – AIBO Revolutions (UNSW 2003 RoboCup Team Thesis),
   University of New South Wales, 2003.
2. Olave, A., Wang, D., Wong, J., Tam, T., Leung, B., Kim, M.S., Brooks, J., Chang, A., Huben, N.V., Sammut, C., Hengst, B.
   The UNSW RoboCup 2002 Legged League Team (UNSW 2002 RoboCup Team Thesis),
   University of New South Wales, 2002.
3. *ImageJ Documentation – Process Menu*
   URL: http://rsb.info.nih.gov/ij/docs/menus/process.html
4. Koehler, K., *Electric Potential*
   URL: http://www.rwc.uc.edu/koehler/biophys/4b.html
5. Robb, G.R.M., *Electric Field Java Demo Program*
   URL: http://local.phys.strath.ac.uk/12-157/lab8/Efield.java
6. Roefer, T., *Quality of ERS-7 Camera Images*
   URL: http://www.informatik.uni-bremen.de/~roefer/ers7/
7. Jensfelt, P., Kristensen, S.,
   Active Global Localisation for a Mobile Robot Using Multiple Hypothesis Tracking
   IEEE Transactions on Robotics and Automation, Oct 2001, *vol 17, no 5, pp 748-760.*
8. Fox, D., Burgard, W., Dellaert, F., Thrun, S.,
   Monte Carlo Localization: Efficient Position Estimation for Mobile Robots
   Proceedings of the National Conference on Artificial Intelligence, 1999.
9. Pham, K., *Project Report,* 2004.
10. Sheh, R., *Visual feature detection for robot soccer*
    University of New South Wales, 2003.

Appendix A

Comparison of unprocessed images and ring corrected images

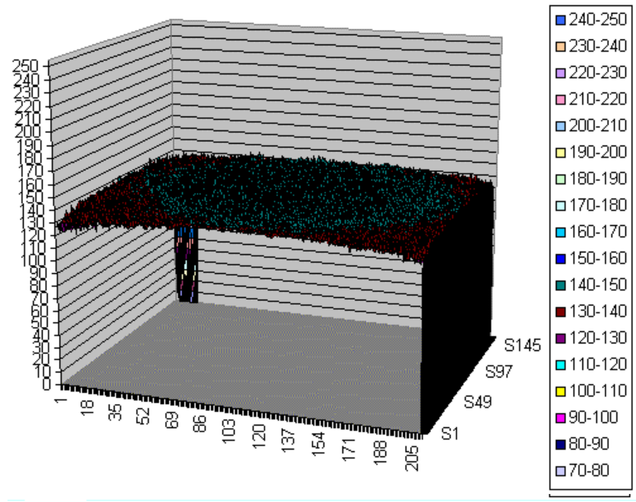
YUV plane of white border – unprocessed

YUV plane of white border – processed

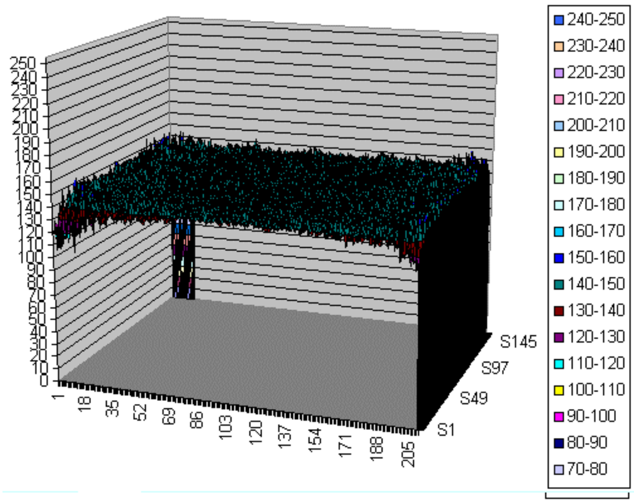


C plane of white border – unprocessed

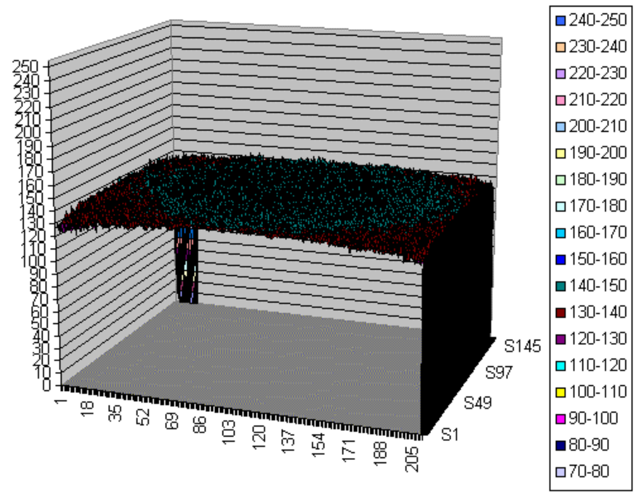
C plane of white border – processed



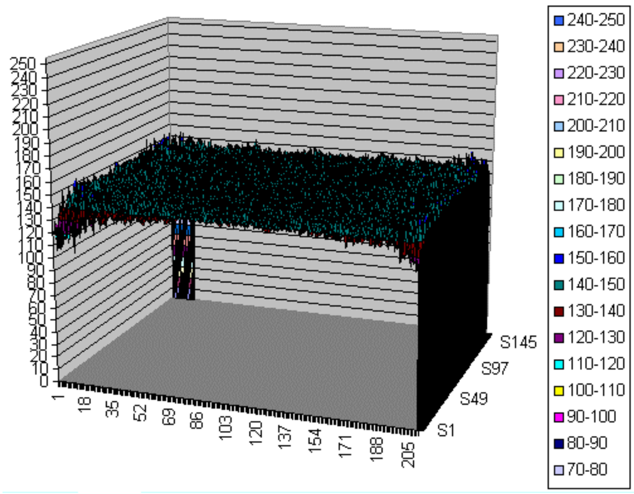
Y plane of white border – unprocessed



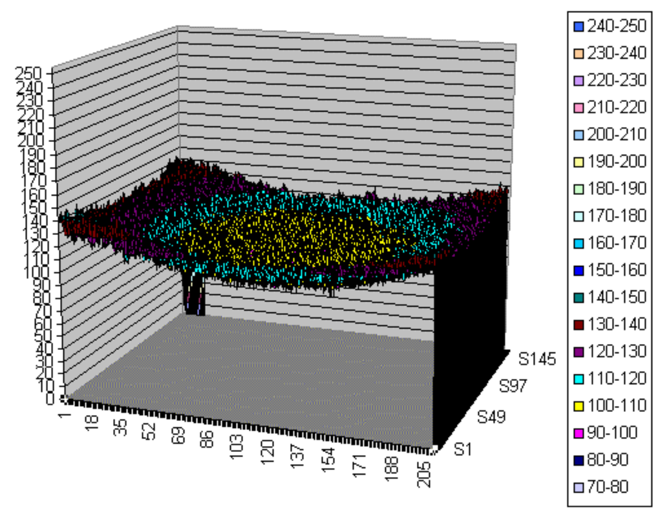
Y plane of white border – processed



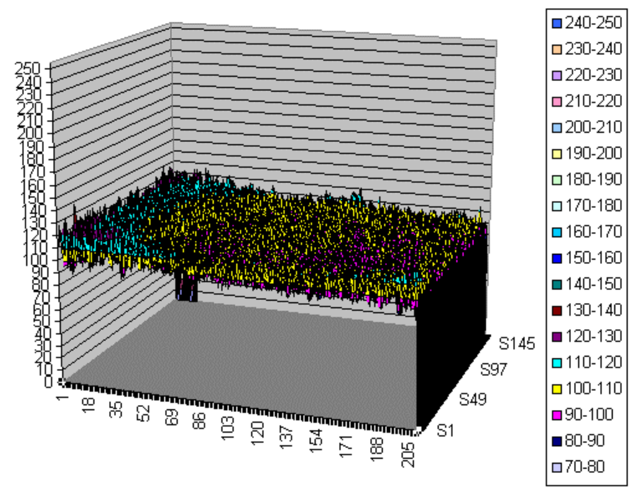
U plane of white border – unprocessed



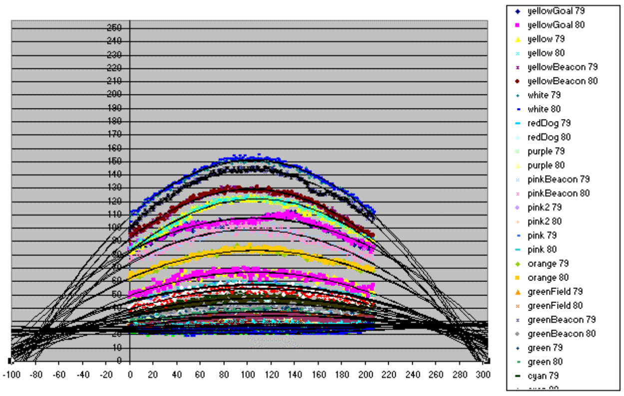
U plane of white border – processed



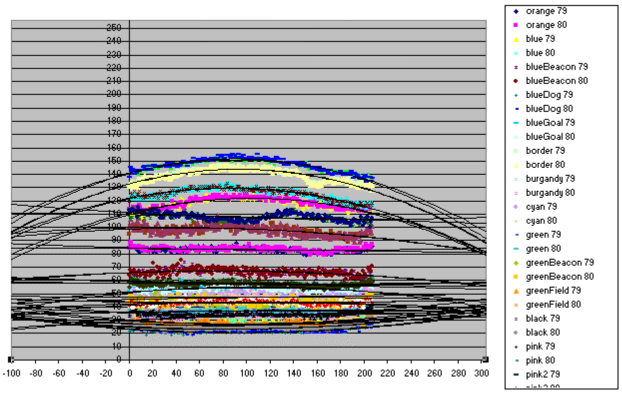
V plane of white border – unprocessed



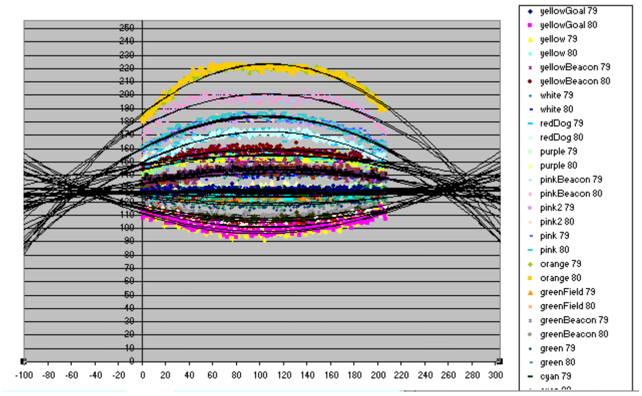
V plane of white border – processed



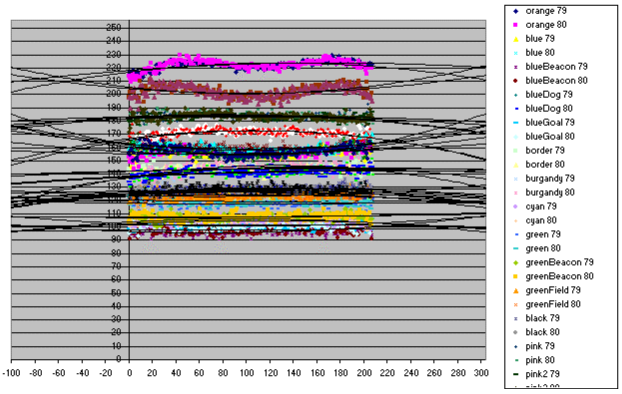
Y plane of uniform colours at Y=79 & Y=80 – unprocessed



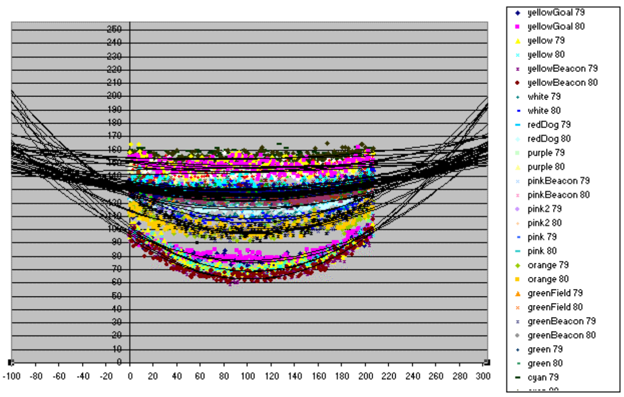
Y plane of uniform colours at Y=79 & Y=80 – processed



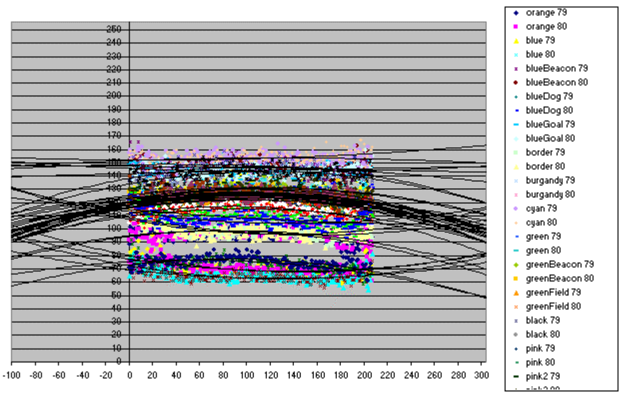
U plane of uniform colours at Y=79 & Y=80 – unprocessed



U plane of uniform colours at Y=79 & Y=80 – processed



V plane of uniform colours at Y=79 & Y=80 – unprocessed



V plane of uniform colours at Y=79 & Y=80 – processed