Proceedings of the 21st IASTED International Conference
APPLIED INFORMATICS
February 10-13, 2003, Innsbruck, Austria

AN ADAPTIVE COLOR SEGMENTATION ALGORITHM FOR SONY LEGGED ROBOTS

Bo Li, Huosheng Hu, Libor Spacek

Department of Computer Science, University of Essex
Wivenhoe Park, Colchester CO4 3SQ, United Kingdom
Email: bli@essex.ac.uk, hhu@essex.ac.uk, spacl@essex.ac.uk

ABSTRACT

This paper presents an adaptive colour segmentation algorithm for Sony legged robots to play a football game. A Self-Organizing Map (SOM) is adopted to measure the current lighting condition and an Artificial Neural Network (ANN) is implemented to produce a suitable General Color Detection (GCD) table. Off-line learning is conducted in color segmentation in order for Sony-legged robots to adapt to lighting changes in the real world. Experimental results show the good performance of the proposed segmentation algorithm.

KEY WORDS

Robotics, Computer Vision, Neural Networks, RoboCup

1. INTRODUCTION

In recent years, autonomous robots have been widely deployed in industry. In most of these applications, the robots need to recognize objects in their environments by vision sensors and navigate their movement through the map dynamically constructed by vision algorithms. However, vision processes can be very complex and time-consuming. Even under an artificial environment, the vision system needs to find the best way for object recognition and good tracking algorithms for moving objects. Real-time requirements normally prevent complex image processing and long-term self-learning. To simplify the recognition process, both color-based and shape-based environments are widely adopted in many applications.

Colour segmentation, as the name implies, uses colour to classify image areas as different objects. Although the goal of a vision system is to reconstruct the whole external environment, especially the objects, the first goal is to reliably locate the objects and beacons in images. In order to find the objects in a colour-based environment, we need to separate different colour areas and find the colour blobs that usually mark the objects. The blob positions can be used later by tracking modules. Basic colour segmentation sets thresholds for each significant color in the color space. Lighting conditions play a key role in the colour segmentation process. Kulessa et al presented a luminance detection method to improve the algorithm performance [9]. Schröter presented his LUT (Look Up Table) method to speed the process [10]. Wyeth et al used the GCD (Generalized Color Detection) method for fast segmentation [15]. Saber et al presented a method that combines both edge and color information for segmentation [10]. Batavia et al used some segmentation algorithms in a real-time system for obstacle detection [1].

In recent years, adaptive colour segmentation algorithms have become one of important research issues in many vision applications. With some learning ability, an adaptive algorithm can improve the performance greatly. Cui et al presented a learning based prediction method for segmentation [2][3][4]. Self Organizing Map (SOM) is an unsupervised ANN learning method and is frequently used in pattern recognition applications. There is some previous work applying SOMs to colour segmentation, e.g. Wu et al implemented SOM in his segmentation work [13][14].

This paper presents an adaptive colour segmentation algorithm, which implements Self-Organizing Map (SOM) and supervised learning to make the algorithm robust. The main problem is to detect the colors in a colour-based environment in order to extract the objects' information as accurately as possible. Varying lighting conditions cause difficulties for the colour segmentation: when the lighting conditions change, the pre-selected thresholds fail to work. Therefore an adaptive vision algorithm becomes necessary for robots to adapt to different lighting conditions and to reduce the need of frequent manual adjustments.

The rest of the paper is organized as follows. Section 2 describes Sony legged robots and colour segmentation. In section 3, both the learning algorithms for segmentation and the supervised learning for thresholds construction are presented. SOM for luminance detection is described in Section 4. Section 5 shows the experimental results and the key problems in this adaptive algorithm. Section 6 presents a brief conclusion and the future extensions.

2. SONY ROBOT & COLOR SEGMENTATION

2.1 SONY LEGGED ROBOTS

Sony legged robots are dog-like robots for entertainment. They are also adopted in the RoboCup soccer competition and used for research on multi-agent systems [7]. Figure 1 presents the agent architecture developed at Essex for Sony legged robots to play a football game in the RoboCup competition [6][8].

The vision sensor embedded in each Sony legged robot is a single PAL camera, with a digital output of up to 176 by 144 pixels. In normal cases, the on-board CPU may only be able to process about 10 frame/second (with an image size of 88 by 72 pixels). Therefore, the development of fast vision algorithms is necessary and crucial for the football (soccer) competitions.

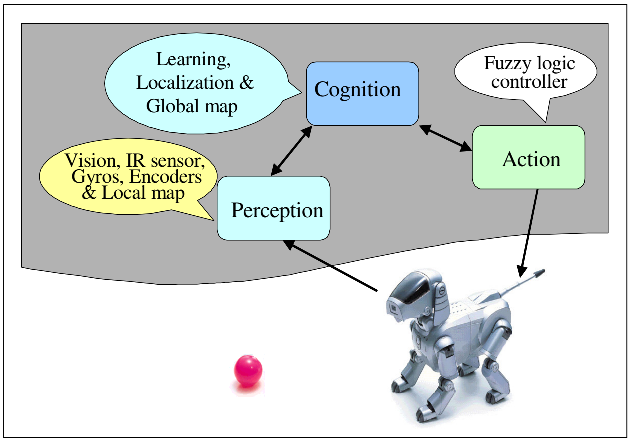

Fig. 1 Agent architecture for the Essex team

The camera can be automatically adjusted in several ways: shutter, aperture, and amplifying rate. There are only three states to be set for each adjustment. Dynamic lighting conditions can affect the image quality greatly. The vision system we developed for Sony-legged Robots consists of two main parts: a colour segmentation algorithm and a tracking algorithm. However, this paper addresses the problem of color segmentation only.

2.2 COLOR SEGMENTATION ALGORITHMS

The task of extracting a color object and estimating its position in images depends on colour segmentation, which needs to be robust to the varying lighting conditions and adapt to dynamic changes in the environment. The first step is to adjust the camera's shutter, aperture, and amplifying rate settings, which is difficult for the Sony-legged robot. Such adjustments may cause other problems, such as change of the depth of view in the image (produced by aperture changes). Therefore the implementation of different thresholds in different lighting conditions becomes necessary. There are three main parts in a color segmentation algorithm, i.e. thresholds under different lighting conditions, lighting condition detection, and the combination of both. Figure 2 shows the relationship of these parts.

The experimental platform is based on a PC and a real Sony legged robot [5], as shown in Figure 3. Some of the above parts, specifically threshold training, lighting metering training, and threshold construction training, are run on a PC for faster learning performance. To control the robot the instructions are sent from a PC to the Sony robot via a serial cable, including moving, capturing images, adjusting camera settings and so on. Through the connection with the robot, the PC can get the images under different conditions. A meter is used for static light measuring, but the result is only used as a reference.

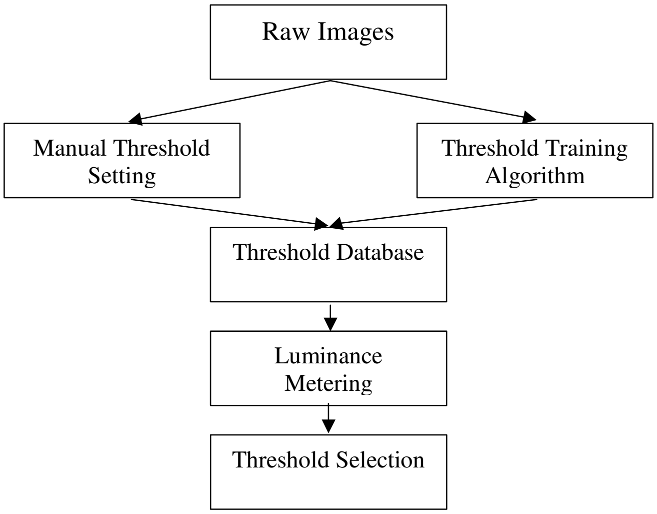

Fig.2 Block diagram of the color segmentation algorithm

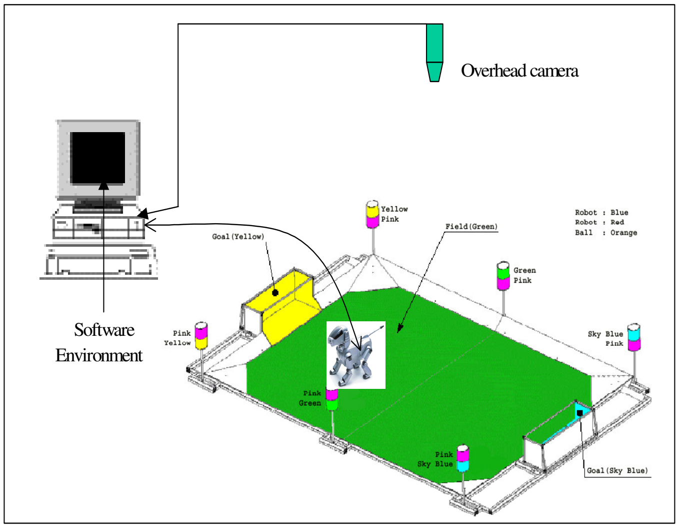

Fig.3 Block diagram of the experimental platform

3. THRESHOLD TRAINING

The thresholds for colour segmentation depend on the colour space in which the operation is implemented. Commonly, a HSV colour space is suitable for colour segmentation. With a powerful CPU or special hardware for conversion between colour spaces, thresholds working in HSV colour space can perform colour segmentation and recognition well. There are many papers discussing colour segmentation in the HSV colour space, with adaptive algorithms or non-adaptive ones [10][14]. However, the camera in the Sony legged robot outputs images in a YUV format and it has a hardware colour segmentation method, namely Colour Detection Table (CDT). CDT conditions can be described as follows:

$$i = P_Y / 8$$
$$T_{U\min i} < P_U < T_{U\max i}$$
$$T_{V\min i} < P_V < T_{V\max i}$$

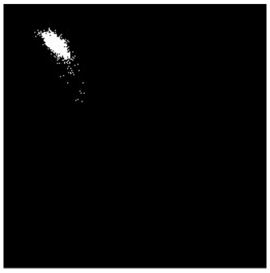

Fig. 4 A plane of a GCD table

There is a rectangular threshold in 2D UV space. It is unavoidable that some unwanted pixels may satisfy such a condition and some wanted pixels might be missed out. The GCD method was presented to solve such a problem [15]. It decides if a pixel belongs to a colour. The GCD condition can be described as:

$$if\left(GCD\left(P_Y, P_U, P_V\right) == colourMark\right)$$

where $\left(P_Y, P_U, P_V\right)$ is in the colour indicated by a colour Mark. It is very similar to the LUT method presented by Schröter [8], which is mostly based on a statistical approach.

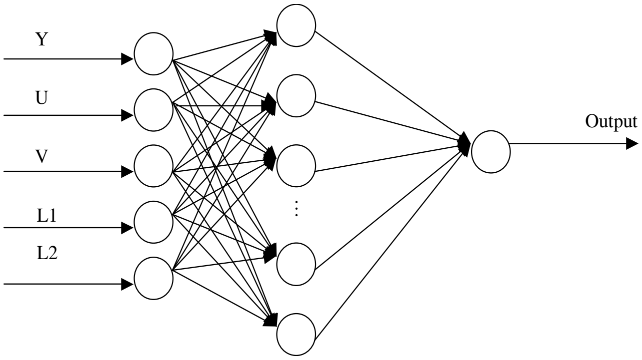

Figure 5 A learning network for GCD training

Figure 4 shows a plane of the GCD table. The thresholds of the GCD method can be any shape in the UV space. The main difficulty with the GCD method lies in how to construct thresholds under different lighting conditions. With a precise threshold in a 2D UV space, it may be very sensitive to the lighting conditions; even a slight light change can make the recognition of a specific colour very difficult. In this case, a training algorithm can help the production of thresholds. With several GCD tables under metered lighting conditions, the program can produce the GCD tables when lighting conditions have changed. Figure 5 shows the learning network used for GCD training.

In this network, Y, U, and V are inputs, and output is the colourmark of the GCD. L1 is the luminance decided by the meter, and L2 is the luminance measured from the image. Training data are current GCD tables that are manually constructed with a set of lighting conditions.

There are many different methods to measure the luminance of the images. Basically, most methods need a patterned object for measuring, which may be not reliable in the application of Sony legged robots. The field colour of a football pitch is green, which is not sensitive to the change of the lighting condition. However, the colour of other objects such as ball and goal is affected by light refection. With different angles, the measured results can vary dramatically under same lighting condition. The metering method based on many images will be described latter.

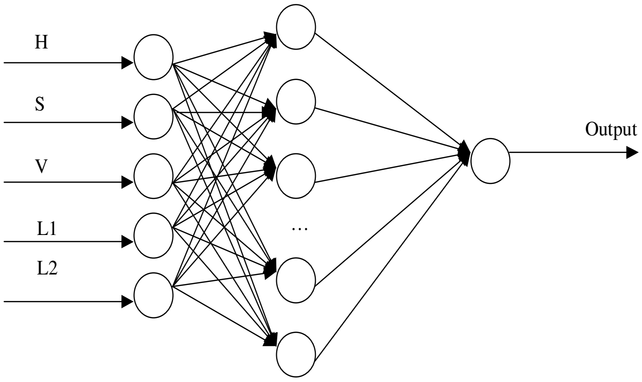

Figure 6 A training network in the HSV colour space

Another problem in GCD table training is the approximation of the wanted GCD and learning results. After a training session with the GCD tables, the result may still be very poor. The results from a simple training method based on a HSV colour space can be easily converted into the YUV with only a small deviation. Figure 6 shows a training network in a HSV colour space, in which three inputs (H, S, V) are different from those in Figure 5.

The output is in a bitmap format. But the shape seems much closer to a rectangle, which makes a simple threshold possible. To convert the HSV thresholds to a YUV GCD table, the following testing is necessary:

$$H(Y,U,V) \subseteq [H_{\min}, H_{\max}]$$
$$S(Y,U,V) \subseteq [S_{\min}, S_{\max}]$$

If satisfied, (Y, U, V) belongs to a colourmark with a threshold as shown above.

4. SOM LUMINANCE METERING

As described above, measuring luminance from images is very complex, especially if there is no patterned object. A dark environment can make some colour look bright, and a bright environment can make something look dark. It depends on many aspects, not only the luminance itself. The key problem is how to produce the powerful colour segmentation algorithms that can detect the current luminance in the processed images and adjust the thresholds automatically.

A real-time vision system is different from a common image processing system. An active vision system can even select an object from the images with several degrees of freedom. With many images and prior knowledge, it is possible for a vision system to detect the luminance in images. One of the problems in many clustering algorithms is that the number of clusters should be specified in advance. The success of a clustering algorithm depends on the specified number of clusters. It is the same case in a basic SOM algorithm.

The more output neurons the higher the resolution, since output neurons correspond to clusters. Different number of clusters leads to different results of tessellation in a pattern space. If fewer neurons are used, lower density data will be dominated by the patterns of higher density data. On the other hand, if more nodes are used, the ordered mapping is difficult to obtain. From a prior knowledge, it is possible to divide the lighting condition to limited levels. Each level corresponds to a GCD table, so that colour segmentation can be very adaptive. The algorithm for the SOM is described as follows:

- *Set the number of nodes N to 2 initially, based on two clusters, and randomly initialize the weights*

  $$w_i = w_i(0), \qquad i = \{1,2\}$$

  where $w_i(k)$ *represents the weight vector of the ith node at the kth iteration.*
- *Draw input x from a training sample set to the SOM randomly.*
- *Find a winner among the nodes using output value.*
- *If (winner!=NULL), adjust the weights of the winner node c and its two neighborhood nodes c-1 and c+1.*

  $$\begin{aligned} w_c(k+1) &= w_c(k) + \eta(k)(x - w_c(k)) \\ w_{c-1}(k+1) &= w_{c-1}(k) + \eta(k)\alpha(k)(x - w_{c-1}(k)) \\ w_{c+1}(k+1) &= w_{c+1}(k) + \eta(k)\alpha(k)(x - w_{c+1}(k)) \end{aligned}$$

  where $\eta(k)$ *is the step size of learning,* $\alpha(k)$ *is a neighborhood function, k is the counter of iteration.*
- *If there is no winner, grow a new node n according to the growing scheme.*

  $$w_n(k+1) = x \text{ and } N = N + 1.$$
- *If a node is rarely win, delete it according to a pruning scheme, N = N -1.*
- *Calculate the distance between each two nodes and perform a merging scheme.*

In a SOM, nodes are statistical results of each image, concerning some colour luminance and vectors like $(l_1, l_2, \ldots, l_m)$. For some pixels of each colour (pixels with the fixed range H value and moderate luminance), the luminance values can be summed to l_i. With a SOM, images can be divided into many groups. While the typical images are mixed with the samples, it is possible for the system to recognize the luminance of images.

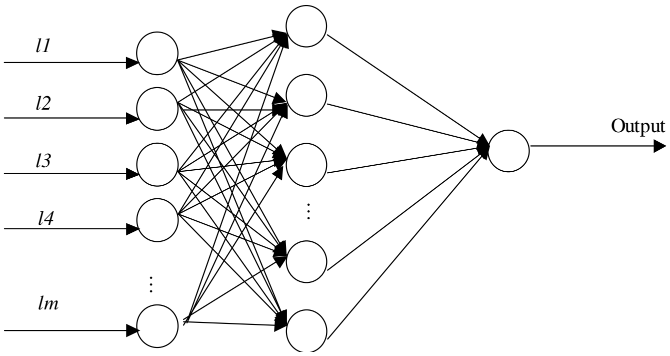

Figure 7 A supervised network

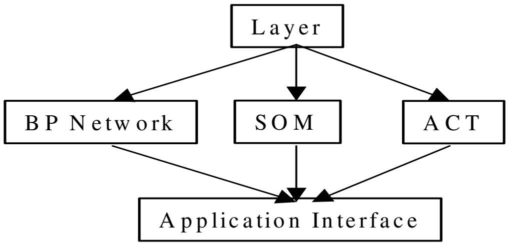

Figure 8 Diagram of the class hierarchy of the ANN

The relationship of the luminance and luminance vectors can be deduced using the result, when a supervised network has been trained by all grouped images mentioned above. Figure 7 shows a supervised network for this purpose.

The output of this network is the luminance level, which can be used next to decide the thresholds for colour segmentation. There are a number of GCD tables available for threshold selection. To make the process more accurate, it is helpful to make a little adjustment of the selected table.

5. EXPERIMENTAL RESULTS

Experiments were carried out using the real Sony robots to show the performance of the algorithm. Three different lighting conditions were adopted in the experiments: 405 Lux, 455 Lux, 535 Lux. GCD tables under 405 and 535 Lux were constructed manually. GCD tables for 455 Lux were constructed by the supervised learning. With large data sets, the speed of learning is very slow. For learning process in a HSV colour space, the network was trained by 32MBytes test data over many hours. The training process in a YUV colour space is faster, but the parameters are difficult to set.

The experiment of training in a YUV colour space produced nothing useful. With the parameters such as the learning speed 5, 1 hidden layer, and 32 hidden neurons, it produced an inadequate GCD table with two many binary values 0 and the learning progress was too slow. On the other hand, with the parameters: learning speed 50, 1 hidden layer, and 8 hidden neurons, it produced a GCD table with too many binary values 1. The learning speed seemed too fast.

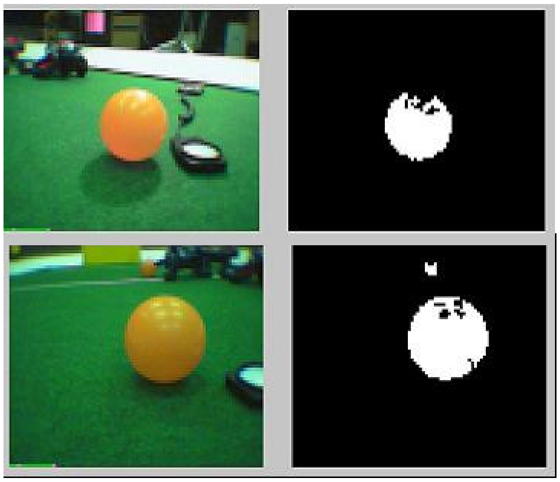

Figure 9 Images processed by suitable GCD tables

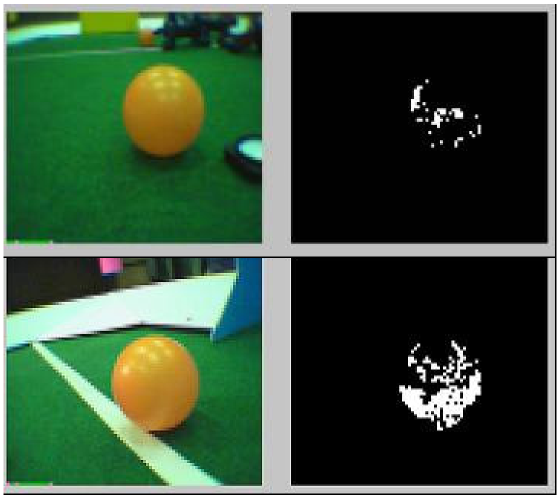

Figure 10 Images processed by wrong GCD

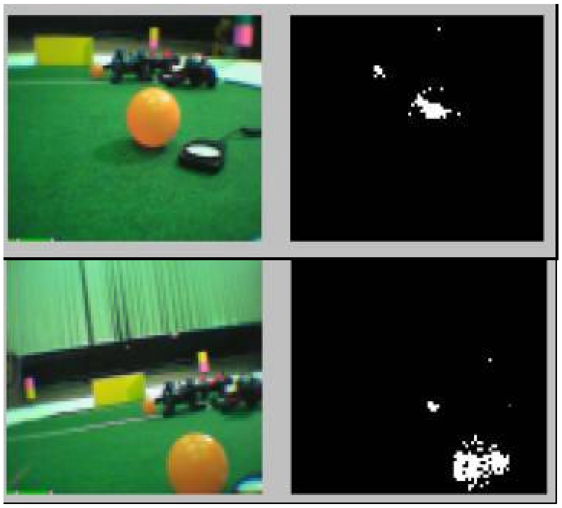

Figure 11 Images processed by a GCD table program

In fact, the result produced by the network in a HSV colour space is also not good enough, although it seems to be approximating the original GCD table very well. When the new GCD is far from the manual one, it needs to be trained with more data in different lighting conditions.

The luminance detection program worked well. It divided all the images into luminance 3 as they were captured. When there is a manual GCD table for each detected lighting condition, the adaptability can be very good.

Figure 8 shows the diagram of the class hierarchy of the neural network used in the experiment. Figure 9 shows images processed by the suitable GCD. Figure 10 shows unsuitable GCD processed images. Figure 11 shows the images processed by the program constructed (adapted) GCD. Figure 12 shows the learning performance of the proposed training algorithm. Finally, the performance of auto Luminance detection under manual GCD condition is presented in Figure 13.

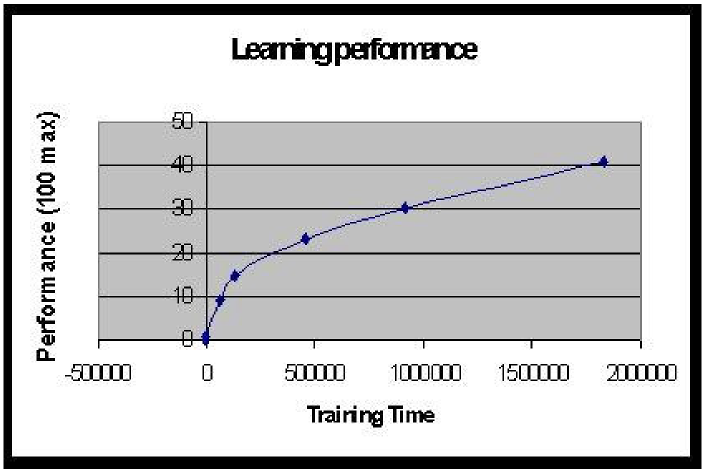

Figure 12 Learning performance of Training algorithm

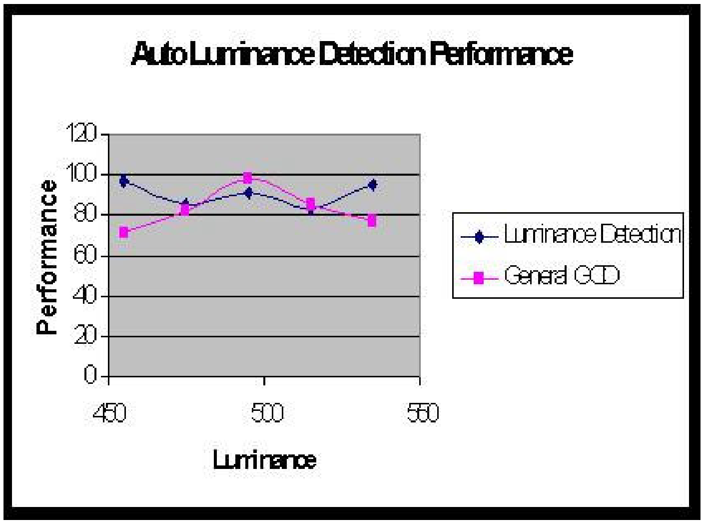

Figure 13 Auto Luminance Detection Performance (GCD by Manual)

6. CONCLUSIONS AND FUTURE WORK

This paper introduces an adaptive colour segmentation algorithm for Sony-legged robots. It works well under different lighting conditions, and is much more robust than a single GCD table in terms of supporting multi-lighting conditions.

For the colour segmentation part, the problem remains: manual operation for the threshold training is time-consuming even for the current half-automatic system. It is possible to implement the shape-based segmentation and SOM for the totally automatic colour detection. We will work on this problem, as well as a robust tracking system for the Sony-legged robot in the next stage of research.

REFERENCES

[1] P. Batavia and S.Singh, "Obstacle Detection Using Color Segmentation and Color Stereo Homography," *Proceedings of IEEE International Conference on Robotics and Automation*, Seoul, Korea, 2001

[2] Y. Cui and J. Weng, “A Learning-Based Prediction-and-Verification Segmentation Scheme for Hand Sign Image Sequences”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.21, No.8, pp. 798-804, 1999

[3] Y. Cui and J. Weng, “Appearance-Based Hand Sign Recognition from Intensity Image Sequences”, *Computer Vision and Image Understanding*, vol. 78, pp. 157-176, 2000

[4] Y. Cui and J. Weng, “View-Based Hand Segmentation and Hand-Sequence Recognition with Complex Backgrounds”, *Proc. 13th Int. Conf. on Pattern Recognitio*n, Vol. III, pp. 617-621, 1996

[5] D. Golubovic and H. Hu, A Hybrid Evolutionary Algorithm for Gait Generation of Sony Legged Robots, Proceedings of the 28th Annual Conference of the IEEE Industrial Electronics Society, Sevilla, Spain, November 5 - 8 2002

[6] D. Gu and H. Hu, “Evolving Fuzzy Logic Controllers for Sony Legged Robots”, *Lecture Notes on Artificial Intelligence 2377*: RoboCup 2001, Editors A. Birk, S. Coradeschi and S. Tadokoro, pp. 356-361, 2001

[7] H. Hu and D. Gu, “A Multi-Agent System for Cooperative Quadruped Walking Robots”, Proceedings of *IASTED International Conference on Robotics and Applications (RA 2000)*, Honolulu, Hawaii, USA, August 14-16, 2000

[8] H. Hu and D. Gu, Reactive Behaviours and Agent Architecture for Sony Legged Robots to Play Football, *International Journal of Industrial Robot*, Vol. 28, No. 1, ISSN 0143-991X, pp. 45-53, January 2001

[9] T. Kulessa, M. Hoch, “Efficient Color Segmentation under Varying Illumination Conditions”, *Proceedings of the 10th IEEE Image and Multidimensional Digital Signal Processing (IMDSP) Workshop*, July 12-16, 1998

[10] T. Rendas and J. Folcher, “Image Segmentation by Unsupervised Adaptive Clustering in the Distribution Space for AUV guidance along sea-bed boundaries using Vision”, Proceedings of *OCEANS*, Hawai, USA, 2001

[11] E. Saber, A. M. Tekalp and G. Bozdagi, “Fusion of color and edge information for improved segmentation and linking”, *Journal of Image and Vision Computing,Vol10,*. pp. 769-780, 1997

[12] S. Schröter, “Automatic Calibration of Lookup-Tables for Color Image Segmentation”, *Proceedings of 3D Image Analysis and Synthesis*, pp.123-129, 1997

[13] Y. Wu, Q. Liu and T. Huang, “Robust Real-time Human Hand Localization by Self-Organizing Color Segmentation”, *Proceedings of ICCV99 Workshop*, pp.161-166, RATFGRTS, 1999

[14] Y. Wu, Q. Liu and T. Huang, “An Adaptive Self-Organizing Color Segmentation Algorithm with Application to Robust Real-time Human Hand Localization”, *Proceeding of Asian Conference on Computer Vision*, *pp.1106-1111,* Taiwan, 2000

[15] G. Wyeth and B. Brown, “Robust Adaptive Vision for Robot Soccer”, “Mechatronics and Machine Vision in Practice”, pp. 41 – 48, 2000