ACSIJ Advances in Computer Science: an International Journal, Vol. 3, Issue 5, No.11 , September 2014
ISSN : 2322-5157
www.ACSIJ.org

Adaptive Color Mapping for NAO Robot Using Neural Network

Vahid Rahmani [1], Vahid Rostami [2]

[1] **Department of Electrical, IT and Computer sciences ,Qazvin Branch,Islamic Azad University, Qazvin, Iran**
V.Rahmani@Qiau.ac.ir

[2] **Department of Electrical, IT and Computer sciences ,Qazvin Branch,Islamic Azad University, Qazvin, Iran**
Vh_Rostami@yahoo.com

Abstract

While playing soccer, the main task of the robot vision system is identifying and tracking objects such as ball, goals, teammate robots and opponent robots. The basis of many object identification methods, particularly those in soccer robots and RoboCup environment, is using algorithms based on pixel color properties. One of the major problems of these robots in RoboCup environment is changes in lighting conditions of match environment and in turn difficulty in identification of environment colors and objects in order to segment the image and identify the objects. In this paper, a pixel color-base identification method has been suggested using a neural network for recognizing the pixels related to each object. The neural network used in this study has 6 output neurons for identifying 6 classes of signs including ball, goal, field, field lines, teammate robot and opponent robot. This proposed method has been tested on over 1000 frames of images received by robot's camera and different data sets and has revealed an appropriate performance with identification rate of over 90% and error rate of 4%.

Keywords: humanoid robot, Standard Platform League, vision system, neural network

1. Introduction

RoboCup, "Robot Soccer World Cup", is an international project to promote artificial intelligence, robotics and other related areas. The RoboCup Standard Platform League (SPL) is one of several active soccer leagues in RoboCup competitions in which the humanoid NAO is used. As the name indicates, in RoboCup Standard Platform League, all robots and rules are identical for all participating teams as a fixed and standard platform. In Standard Platform League [1], all participating teams are only allowed to compete using the humanoid NAO. In this RoboCup league, robots play soccer fully autonomously and without interference of human. In these competitions held annually by international committee of roboCup, rules, regulations and conditions of competition environment are identical for all participating teams [2]. According to these rules, pre-determined color and size of all objects on the field are described. For example, the ball is red, the goals are yellow, the teammate robot clothing is red and opponent robot clothing is blue, and these colors will be fixed throughout the whole match. While playing soccer, the main task of the robot vision system is identifying and tracking objects such as ball, goals, teammate robot and opponent robot. The basis of many object identification methods, particularly those in soccer robots and RoboCup environment, is using algorithms based on pixel color properties. According to the report of RoboCup competitions website, annually an average of 32 teams succeed to gain the permit for participating in these competitions and according to the technical reports submitted to the competitions committee, approximately all these 32 teams use pixel color-based algorithms to identify the objects. According to the technical report [3] in 2012, Austin Villa team has used a linear scanning algorithm [4] to identify objects and creating a color table of known objects on the field to segment the image. Similarly, according to the technical report [5] in 2013, B-Human team from Germany, which is one of the prominent teams in this league, has used a linear scanning algorithm. The robot's camera provides images in YUV442 color space with 640 × 480 resolution. Images are scanned on a series of vertical lines. A color-table is used to determine the colors in the image. The color-table determines that which class each color belongs to. In addition, this team has used the k-nearest neighbor (knn) method based on 3-D tree to accelerate this process. According to technical report [6] in 2013, NTURobotPAL team has used the Belief-merge algorithm to strengthen the system against unstable probable conditions. According to technical report [7] in 2013, SPiTeam has also used ultrasonic sensors and bumper to identify the collisions and linear scanning algorithm and pixels colors for assigning labels to each pixel. According to technical report [8] in 2014, Team-NUST has used the combinatory and probabilistic method based on color and shape of the objects using OpenCV function library of to identify the objects. One of the main challenges of these teams in competitions is regulating, identifying and learning color-tables in robot as this process often encounters difficulties. Since color properties are not resistant to the changes in lighting condition and noise conditions and colors have different properties in different lighting conditions, changes in the lighting condition of the field or appearance of noise face the robot with difficulty in identifying the colors in the changed lighting condition and robot is confused and can no longer recognize the colors correctly given the color-table set in desired and ideal condition before starting the game. Hence, this is referred to as one of the major problems RoboCup teams encounter. Therefore, this study tries to provide a more resistant method compared to linear scanning method to changes in lightning condition using a neural network.

2. Color Space

Color spaces are methods of coding and recognizing colors used in several color combinations. YUV and RGB are two models of color spaces with the most sensitivity. CMOS camera in NAO robot captures images in bit map of YUV format with 640×480 pixels in vision module, and frames are processed at the rate of 30 frames per second. Each point in YUV color space corresponds to a certain color. A color class is a set of all YUV colors which can be observed in pixels like the object with a certain color. In other words, each color class is a subset of color space including all changes of a certain color representing real world attractiveness. All colors are classified based on the required application. Fig. 1 shows an example of colors existing on the field in different lighting conditions and Table 1 shows examples of YUV codes of colors used on the field. Fig. 2 represents the color range in YUV space and the degree of distinction of colors in relation to each other. As seen in Fig. 2, the colors of all 6 groups of objects in YUV space are completely separate and distinguishable.

Table 1: Respective objects colors in YUV for the colors in Fig. 1

(YUV)	(YUV)	(YUV)
(81,126,69)	(61,136,77)	(32,109,182)
(81,126,70)	(166,103,39)	(22,115,165)
(78,125,71)	(233,61,137)	(54,117,209)
(99,133,65)	(177,27,174)	(55,113,212)
(96,129,59)	(219,22,146)	(108,144,57)
(68,116,78)	(189,59,166)	(153,142,77)
(112,134,68)	(149,55,145)	(32,109,182)

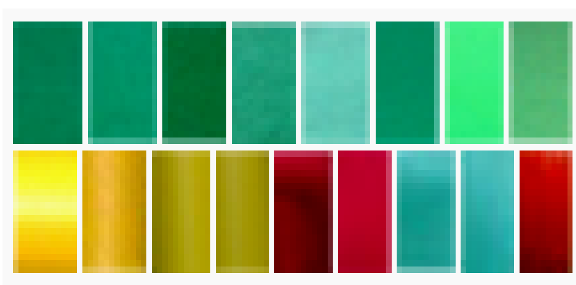


Fig. 1 Typical object colors from different illumination condition.

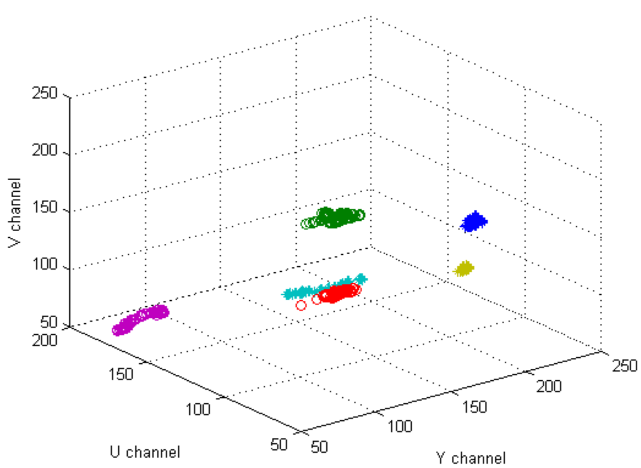


Fig. 2 object color sample plotted in 3-D YUV color space .

3. Training Samples Dataset

Collecting enough suitable training data is necessary to implement the identification system and achieve an acceptable result. To this aim, a set of images were captured in different lightning condition, different intervals, different positions and angles (rotation) and in noise conditions from 6 groups of objects by NAO robot's camera. Fig. 3 shows an example of used dataset.

Fig. 3 sample of the images used in the training phase in different illumination condition.

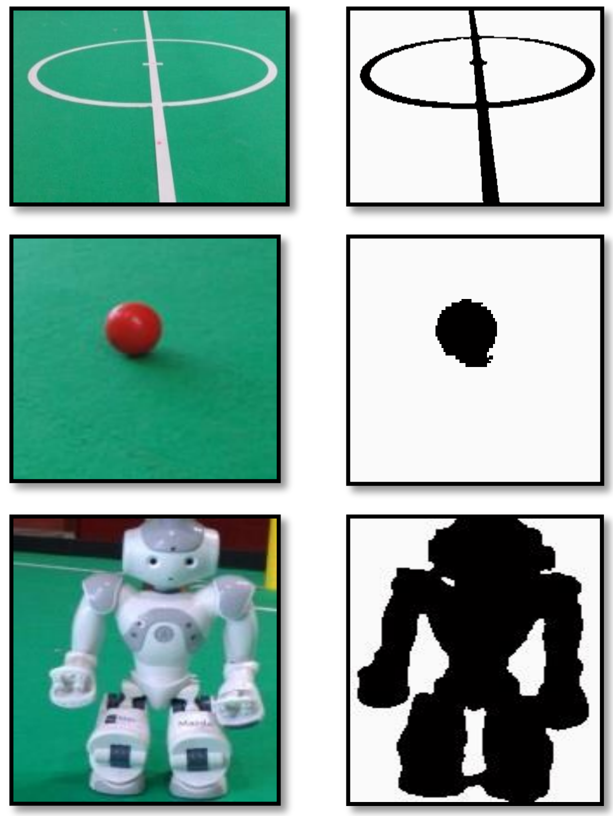
4. Neural Network

Neural networks with their considerable capability in achieving results from complicated and ambiguous data can be used in deriving patterns and identifying different trends whose identification is very difficult for humans and computers. The task of the robot's vision system is identification of pre-known objects such as ball, goals, field, lines, teammate robot and opponent robot. To identify pixels colors and their classification, a multi-layer perceptron (MLP) neural network has been employed. The number of input neurons in this network is equal to the number of YUV color space channels, that is, 3 neurons; and the number of output neurons equals that of the object types on the field, that is, 6 neurons. Thus, each pixel belongs to an object out of 6 objects of ball, goal, field, field lines, teammate robot and opponent robot. Hidden layer which consists of processing neurons is the place for data processing. The number of layers and neurons in each hidden layer is usually determined by trial and error method. The implemented network here has one hidden layer with 5 neurons. Network training is through back propagation method and activation function for hidden layer neurons is of Log-sigmoid type and of tangent sigmoid type for output layer neurons [9]. For training the network, 1000 training data as shown in Table 1 have been used as network inputs. In test phase, YUV color value of each pixel for each input image is applied to the network and the network output is calculated. The neuron whose maximum output value is higher than a certain threshold value determines the class type of pixel related to the input. If maximum output of neurons is lower than the set threshold, the system will announce that pixel as unknown and classify it into the unknown pixels category. This threshold value has been considered differently for each group of objects. The following table shows the threshold values considered for each group.

Table 2: considered threshold for each group.

objects	ball	goal	Teammate	opponent	field	line
Threshold	0.6	0.8	0.9	0.9	0.7	0.6

After classifying all pixels of the input image, a segmentation algorithm with an appropriate threshold value is used to segment the image for displaying the objects, so that the colors of all pixels of an object in each group is replaced with a unique color. Fig. 4 shows an example of segmentation.



(a) (b)

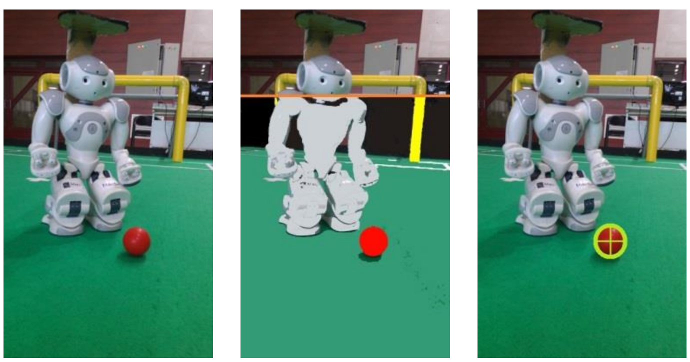
Fig. 4 sample of the images segmentation. (a) raw images and (b) segmented image

5. Results

To decrease the calculation load and increase the processing speed, the images received from the camera are not processed completely, but a point in a far distance from the point of robot's camera is projected in the horizon and a line along the image is drawn in both sides from that projected point along the horizontal vector. Since all objects are on the field, all pixels above the horizon line are left unprocessed and only those pixels below the horizon line are processed [10]. An example of pixel processing and identifying the ball is shown in Fig. 5. To evaluate the network performance, false positive rate (FPR) evaluation factor and accurate detection rate (DR) have been used for each group of objects according to the following relations:

$$FPR(i) = \frac{No.of\ non - member\ pixels\ classified\ as\ group\ (i)}{total\ no.of\ non - member\ pixels} \quad (1)$$

$$DR(i) = \frac{No.of\ group(i)\ pixels\ correctly\ classified}{total\ no.of\ group(i)\ pixels} \quad (2)$$



(a) (b) (c)

Fig.5. sample of the images segmentation and Ball detection. (a) raw image (b) segmented image and horizon line (c) Ball detection

Evaluation factors have been calculated separately for each group of objects. The results indicate that the suggested method shows high success and accuracy percentage. Table 3 shows the results of the suggested method evaluation.

Table 3: Evaluation FPR and detection rate (DR) factors.

Object	DR (%)	FPR(%)
Ball	96.33	6.21
Goal	95.65	9.45
Robot	98.83	3.72
Line	90.08	11.63
Field	97.83	5.67

In addition, to evaluate the accuracy level of our suggested method compared to the linear scanning method (introduced in the introduction), both were tested in five different lighting conditions.

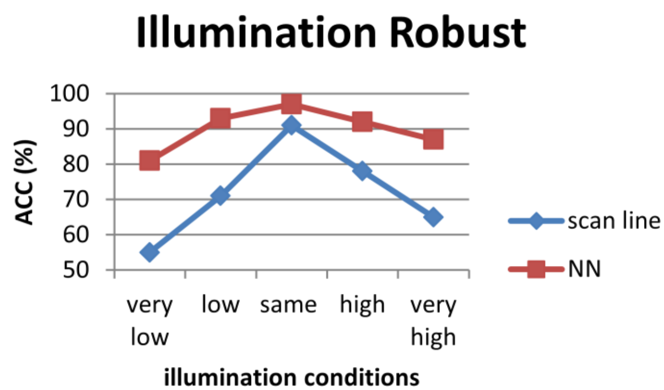


Diagram 1: shows the performances of both methods in different lighting conditions

The results show that the proposed method has a better performance compared to the linear scanning method in terms of accuracy and resistance to changes in lighting condition. Diagram 1 shows the performances of both methods in different lighting conditions.

6. Conclusions

This paper presents a real-time auto-adjusting vision system for robotic soccer. During testing the system seems to be able to meet its requirements. The robots were able to localize themselves on the field and to handle the ball, goal, line, robot and other landmarks in an accurate manner. Future research will try to extend the approach in a way that allows the automatic object detection without color table and remove color table process, because colors adjustment on color table in matches take a lot time and memory space.

References

[1] RoboCup SPL, “The RoboCup Standard Platform League,” 2014, http://www.tzi.de/spl

[2] RoboCup Technical Committee. RoboCup Standard Platform League (Nao) rule book, 2014- 05-08. available online: http://www.tzi.de/spl/pub/Website/Downloads/Rules2014.pdf.

[3] S. Barrett, K. Genter, M. Hausknecht, T. Hester, P. Khandelwal, J. Lee, M. Quinlan ,A. Tian, P. Stone, andM. Sridharan. Austin Villa 2010 standard platform team report. Technical Report UT-AI-TR-11-01

[4] Matthias Jungel, Jan Hoffmann, Martin Lotzsch. A Real-Time Auto-Adjusting Vision System for Robotic Soccer

[5] Rofer, T., Laue, T., et. al B-Human Team Report and Code Release 2010, http://www.bhuman.de/file_download/33/bhuman10_coderelease.pdf.

[6] Chieh-Chih Wang, Chun-Hua Chang, et al. NTU RoboPAL Team Description for RoboCup 2013

[7] SPiTeam- Applications for Participation RoboCup The Netherlands 2013 -Team Description Paper for RoboCup 2012-www.spiteam.org

[8] Dr. Yasar Ayaz, Sajid Gul Khawaja, Team-NUST Team Description for RoboCup-SPL 2014 in João Pessoa, Brazil

[9] L. Fausett, Fundamentals of Neural Networks, Prentice-Hall, New Jersey, 1994.

[10] Ehsan Hashemi, Maani Ghaffari Jadidi, et. **al** MRL-SPL Code Release and Team Report 2012**.**