

Shadow Compensation for Computer Vision in a Robot Soccer Team

Sorawish Dhanapanichkul Prabhas Chongstitvatana

Department of computer engineering
Chulalongkorn University, Bangkok, Thailand
E-mail: sorawish@gmail.com, prabhas@chula.ac.th

Abstract - This paper introduces a method to compensate for shadow in a computer vision system for a robot soccer team. For a non-uniform lighting, a simple color classification based on a fixed threshold will fail. The shadow compensation method uses a lighting model to perform the color classification based on the albedo ratio. The system has been tested on the robot soccer team that participates in Robocup 2005. The system performed well with improving percentage of the correct classification.

I. INTRODUCTION

This paper describes how to use a simple light equation to compensate light that is non-uniform and also describes an overview of a global vision system of a robot soccer team. This system has to be robust to the change in lighting condition that is non-uniform and can vary over the time. The proposed method has been used in our Robot soccer team that participates in Robocup 2005 [1] in Japan. The vision system delivers the positions and orientations of our team's robots and positions of the opponent's robots and the ball.

Robot Soccer

The proposed computer vision system has been use in Robocup (Small size league) in which the event has staged annually. In the competition, each team may has a maximum of 5 robots and each robot's height must not exceed 15 centimeters. Each team will have markers place on the top of robot's pattern. This pattern has two colors, blue and yellow. An orange golf ball is used as the ball in the game. The field area is 4.4 x 5 metres. The rules of the soccer game are adapted from FIFA's rules to be suitable for robots. All systems must be fully autonomous.

II. VISION SYSTEM

This section describes an overview of the vision system. The system is divided into two parts. First part is all about image processing. Second part is a data fusion. It is interfaced to the control system. An AI main system issues all motion commands to the control subsystem. The visual sensors are two Toshiba IK-TF5 analog cameras. The frame grabbers are two matrox meteor II/Multi-channel. The software system operates on two Pentium 4s, 2.4 GHz 512MB processors and it operates at 60 frames/second with an image size 640 x 480 pixels. Each image is processed separately until the second part. First step of the process is the undistortion of the image using Tsai's al-

gorithm [2]. Then each pixel will be classified by a look up table into a color index and luminance of that pixel.

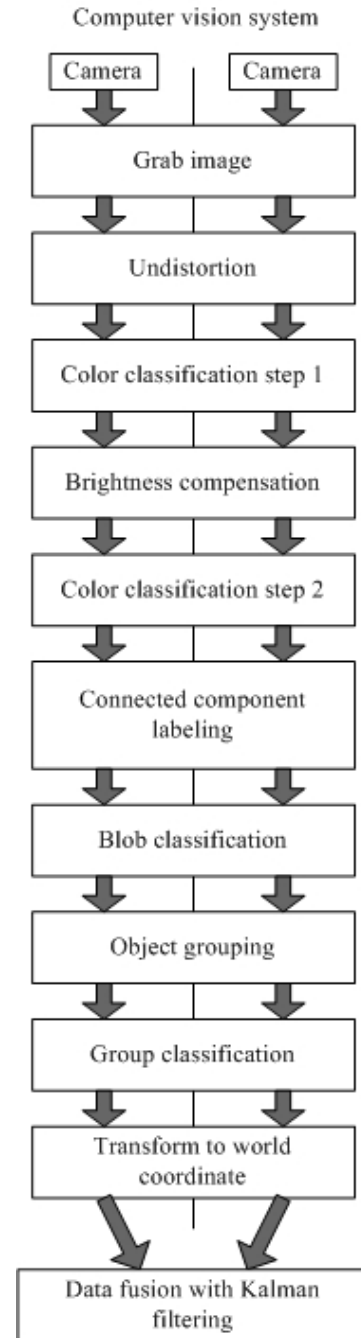


Fig. 1 The diagram of vision system

Next, each pixel's luminance will be compensated and will be classified again into a final color index. When all pixels are classified, the connected component labeling process begins. In the connected component labeling process, each connected group's parameters are calculated. The connected group's parameters are used to define the shape of the group which is divided into two types, the bar and the disc. All shapes are grouped together to make a possible pattern. This pattern is used as an identifier for each robot or a ball. Either a robot is our own team mate or it is an opponent robot. After the classification, all objects' position is corrected by Tsai's algorithm again to find the world coordinate of that object. The second part begins with Kalman filtering to fuse data from two cameras. The input information into the data fusion stage also comes from the commands to the robots issued by the AI system. The next section describes the steps of each process. A detailed explanation is given on how to classify the color of pixels and how to compensate the light. Fig. 2 shows an input image from one camera.



Fig. 2 The input image.

A. Undistortion

This process uses a look up table from a prior calibration to map the correct coordinates with the input pixels. Tsai's algorithm [2] is used to calculate the mapping function.

B. Color classification step 1

Let $P_{x,y}$ be a pixel (x, y) which consists of illumination in three channels r, g and b

IC is an color index

h, s, v be hue, saturation and brightness

$H(P_{x,y}) = h$ returns hue of $P_{x,y}$

$S(P_{x,y}) = s$ returns saturation of $P_{x,y}$

$V(P_{x,y}) = v_{x,y}$ returns brightness of $P_{x,y}$

Let $\{(MaxH_{IC}, MinH_{IC}), (MaxS_{IC}, MinS_{IC})\}$ be the property of color IC

$$T(P_{x,y}) = IC \text{ iff}$$

$$MinH_{IC} < H(P_{x,y}) < MaxH_{IC} \text{ and} \quad (1)$$

$$MinS_{IC} < S(P_{x,y}) < MaxS_{IC}$$

The look up table is created before this process begins. This look up table receives the r, g, b as an input and the output are color index and luminance of that pixel. A color index consists of orange, dark blue, yellow, cyan, pink, white, green, and uninterested color.

C. Compensation of the light

A simple model of image brightness [3] is as follow:

$$v(x) = k_c I(x) \rho(x) \quad (2)$$

v is camera response at point x

k_c is camera gain

I is illumination

ρ is albedo

So if two objects are in the same position and the light intensity is not changed k_c and I will remain the same and the result is:

$$\frac{v_1}{v_2} = \frac{\rho_1(x)}{\rho_2(x)} \quad (3)$$

If we compare between the ratio of robots and the playing field in the same position and the same light condition, we will get the albedo ratio between the field and the color that we are interested (3), Rc_{IC} . To reduce the complexity of the algorithm we divide the playing field into many small areas and find the average intensity of the field in that area, let it be $v_{xa,ya}$. Then we find the reference point that has the maximum light's intensity v_{ref} and find the Rc_{IC} of this area. Then we can calculate the intensity that have to be compensated in each area and each color. The difference intensity is

$$\Delta v_{IC,xa,ya} = \frac{v_{ref} - v_{xa,ya}}{Rc_{IC}} \quad (4)$$

and then

$$v'_{x,y}(P_{(x,y)}, IC) = v_{x,y} + \Delta v_{IC,xa,ya}$$

We will know the corrected intensity of each color in each area from the equation above. Table 1 shows the result of Rc_{IC} from the experiment. Fig. 3 shows the result at the end of this light compensation step.

Table 1 Albedo ratio between field and other color

| Color Index | Rc_{IC} |
|-------------|-----------|
| Orange | 0.3333 |
| Yellow | 0.2778 |
| Dark blue | 0.6667 |
| Green | 0.2778 |
| Cyan | 0.4761 |
| Pink | 0.2778 |
| White | 0.2 |



Fig. 3 The image that has already been compensated

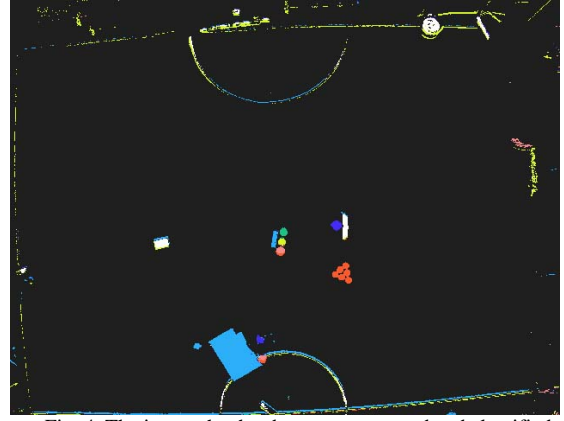


Fig. 4 The image that has been compensated and classified

D. Color classification step 2

While the first table is computed, we compute the second table using the function below. This function (5) computes the color index and corrects the brightness of each pixel and classifies it.

$$T'(IC, v'_{x,y}) = \begin{cases} IC | \text{Min}V_{IC} < v'_{x,y} < \text{Max}V_{IC} \\ C_{unint\ erest} | \begin{cases} v'_{x,y} \geq \text{Max}V_{IC}, \\ v'_{x,y} \leq \text{Min}V_{IC} \end{cases} \end{cases} \quad (5)$$

We setup an environment which has non-uniform light (half of the field does not have a light so the light only comes from the other half). The robot is commanded to move across the half line of the field while the data is being collected. We compare the result from two situations, with and without brightness compensation. Table 2 shows that the correctness in detection of targets improved almost double when using light compensation. Fig. 4 shows the image after the light compensation and the classification in this step.

Table 2 The result from the experiment in percent of detection

| Situation | Percent |
|---------------------------------|---------|
| With brightness compensation | 50.3 |
| Without brightness compensation | 97.1 |

E. Connected component labeling

This process connects the pixels that have the same color index and are spatially connected together to the same component using the method in [4]. These components are called blobs. The parameters within this component are calculated and their labels are assigned.

F. Blob classification

The parameters in the above section are used as the criteria to divide blobs into two types, the disc and the bar. Two most important parameters are the eigen vector and the eigen value. We determine the ratio of the eigen values to determine the type of a blob. If the ratio is less than 5.00 then it is classified as a disc else it is a bar.

G. Grouping and classification of each group

Objects are classified into three types; our robots, the ball and the opponent robots. Each blob will be grouped to others to find a possible pattern. The ball and opponent robots will be detected using only shape and color. Any multiple detection of objects are sent to data fusion and filtering step.

H. Transform to the world coordinate

At first, all of objects' coordinate are in image coordinate. To use them in the control system, a world coordinate is required. All coordinate data are transformed into the world coordinate using Tsai's algorithm.

I. Data filtering and fusion

We divide this step into three parts. First part handles robots of our team. Since we know that each of our robots has a unique pattern for identification, we can identify our robot individually. The robots' translation and orientation data will be filtered by Kalman filtering [5]. Second part handles the ball. There is only one ball in the playing field and we assume that the ball's velocity will not be more than 10 meter/second. This assumption bounds the velocity of the ball that have been detected by two cameras. The data are filtered by Kalman filtering. The third part handles opponent robots. This part is the most complex problem because we do not know how many robot the opponent team has and we do not know the pattern of

the opponent robot. We use a probabilistic matching to locate each opponent robot. The probability of finding an opponent robot depends on the distance from the location where we last find it. We do not apply any velocity filtering to an opponent robot data because this information is not used by our AI system.

In this process, one important step is the prediction step. Because the image being processed has delayed in the system itself. The system delay is about 5 frames (about 100 milliseconds). Fig. 5 shows the pipeline of our process and its delay time. To compensate for this delay, a feedforward prediction data is used in the control system.

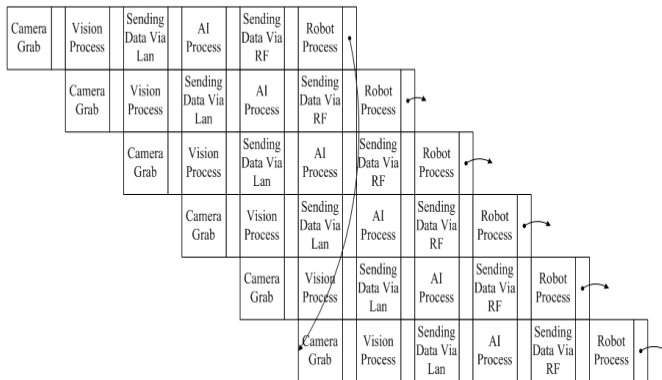


Fig. 5 The delay of the overall system

III. CONCLUSION

This paper introduces a new approach to compensate the non-uniform brightness over the field. The proposed method is capable of classifying the colors even in the non-uniform light. This system can perform in real-time at 60 frames/second. The result from the shadow compensation improves the correctness of classification greatly. The system has been tested during this year Robocup 2005 in Japan. The system worked satisfactorily. Our future work will be concentrated on the following tasks:

- 1). Robustness for color classification in a time varying lighting condition.
- 2). Improving the time efficiency in classification of color. This is quite crucial for a demanding task such as robot soccer competition.

REFERENCES

- [1] Robocup 2005. <http://www.robocup2005.org/home/default.aspx>
- [2] Tsai, Roger Y. (1986) "An Efficient and Accurate Camera Calibration Technique for 3D Machine Vision," Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Miami Beach, FL, 1986, pp. 364-374.
- [3] Forsyth, Ponce. Computer Vision: A MODERN APPROACH. Pearson Education, 2003.
- [4] H. Samet, "Connected component labeling using quadtrees," J. ACM 28(3), pp. 487-501, 1981.
- [5] Gary Welch, Gary Bishop. Introduction to Kalman filter. SIG-GRAPH 2001 Course 8.