The Research of Reducing the System Resources Occupancy Rate of Color Discrimination System in the Flushbonading System
Canjie Liu JIanjun Huang Zixin Liu
Guangdong Experimental High School 2608309161@qq.com

The Research of Reducing the System Resources Occupancy Rate of Color Discrimination System in the Flushbonading System

1.Introduction

The Botball is a international scientific and technological competition. It requires competitors to utilize a lot of scientific products, including abundant sensors Botball offered, which allow the robots to get information from the environment to guide them correctly. In the process, the advanced sensor which needs to match arithmetic with the program to use is camera. However, when using the camera to discriminate colors, a conflict occurs. When using low resolution to distinguish colors, the speed turned out to be fast. But the problem is that the accuracy is not that good, for it sometimes recognizes red ball as green ball. When using high resolution, its accuracy is better though it runs slower. Thus, it will affect the detection of other sensors. After searching information, raising assumptions and analyzing the data, the research is made below.

2. Photographing and color-detecting

The process of identify an object using a camera begins with photographing it and then transport the visual picture into data for system to analyze. During transporting from visual information into mathematical, the unit of the picture is called pixel, and the number of pixel in the picture is called resolution ratio. The higher the resolution ratio is , the larger the number of pixel is. When detection, the larger number of pixel will lead to the more accurate result of detection.

When the visual identification is processing, it will put a frame of picture into the RAM and detect each pixel separately, which means its rate is slow. Only by using powerful CPU and RAM, can it detect the size and the precise position of an object.

If we only use the camera to separate black and white, the procedure will be fast even when the area need to be scaled is large, since only the data of grayscale is detected. However, the price of using camera only to detect the grayscale is large, and the value of the grayscale did not vary much when it comes to different colors, so threshold value is unable to establish. Also, the problem with camera itself, which is extremely slow for detecting, is hard to conquer in the competition since accuracy and speed can not be both satisfied under this circumstances. What's more, the program for running it is very complicated.

But the bright side of using camera is that it can detect larger scales and gather more information, such as the object's size and position. It also removes the limitation done by complex facility, which broaden the area of using.

3.Practical Use of Camera in Color Discrimination

3.1 Disposing Process

Nowadays, the majority of method of VI(visual identity) contains these steps below. First, the controller take in one of the frames which is just taken by the camera. Then it distinguishes and analyses every pixel in the picture. Through distinguishing every color which the pixel represents, it counts how many pixels illustrating the same color totally and concentrating in an area. Once the data is up to a standard that is set before, the processor will define whether the object is exists, what's its color and where it is located. Moreover, this process requires the controller to distinguish every frame, operate the steps below again and again in order to release color discrimination.

Neither the configuration of wallaby controller nor internal storage is low, however, before using new method, the result turned out not to be that good. This is the program used in the process.

```
#include <kipr/botball.h>

int main()
{
   enable_servos();
   camera_open(MED_RES)；
   set_servo_position(0,0);
   msleep(1000);
   while(digital(0)==0)//when the switch is touched ,the program will stop
   {
      camera_update();
      if(get_object_count(0)>0)
      {
         set_servo_position(0,1024);
         msleep(1000);
      }
      set_servo_position(0,0);
      msleep(1000);
    }
  return 0;
  }
```

Actually, the program above is useful in detecting colors; but when it comes to the detection of the ball, it fails. Because when the servo is running, the camera remains open. That's what used a large amount of the system space which leads to the disability to run

correctly.

3.2 Practical Result of Operating

Our team once used the program above to test the camera's ability of color discrimination. It turned out that the servo will twirl for a certain angle. Altering a few minutes, reset the servo and circulate these steps. In our assumption, the servo will twirl immediately when a red ball is set in front of the camera. However, against our thought, in the real test, the servo always takes 10 seconds or so to give a new feedback after it twirls, which implies that even though the program has took up the majority of resource of the processor when using this method of calculating, the operating efficiency is still low. If we lower the distinguish ability of the camera, the reacting speed goes up while the accuracy of distinguishing falls. Using the unclear picture, the processor sometimes makes mistaking when analyzing, such as recognizing the green ball as red ball.

3.3 Problems Appeared during the Operation in the Competition

When a botball robot is competing in the field and VI is needed, other sensors' detections will stop except the color discrimination. That's because using this kind of color discrimination program will take up the majority resource of processor. Meanwhile, it will possibly result in the system halted, the wrong detection and the destruction of the previous program set if continually using other sensors to detect. Thus on the field for competing , a few teams will use VI. These behavior abandoned one of the most important usage of a camera and disobeyed the committee of offering cameras.

4. The First Way of Reducing the Usage of Resource of the System in Color-Detecting Process

4.1 The assumption of the conflict

After discovered the efficiency of processing is low, we assumed that the conflict between color-detecting program and servo-running program caused the Processing unit overloaded, which leads to a result of suspending of the Processing unit and a slow-reaction appearance.

We then deleted part of the program which control the moving of the steer and perform another experiment. The reaction of putting a red ball in front of the camera is a character of speedy feedback, "red." The conclusion is that it is the conflict usage of Processing unit suspend it.

4.2 The Method of Fixing

Based on the former information, we overwrote the program to finish this process: begin color-detecting, red ball detected, end color-detecting, begin servo-running, begin color-detecting. Using this program, the reaction of the servo is prompt comparing to the former version.

The following is the modified program:

```
#include <kipr/botball.h>
```

```
int main()
{
    enable_servos();
    set_servo_position(0,0);
    msleep(1000);
    while(digital(0)==0)//when the switch is touched ,the program will stop
    {
        camera_open(MED_RES)；
         camera_update();
         if(get_object_count(0)>0)
         {
            camera_close ()；
               set_servo_position(0,1024);
               msleep(1000);
         }
         set_servo_position(0,0);
         msleep(1000);
      }
   return 0;
   }
```

Before the servo moves, we first close the camera in order to save more space for running the next program.

From the former discussion, we conclude the first method of solving the problem: end or begin the program of color-detecting and reduce the conflict between other program in order to accelerate the process of color-detecting and improve the efficiency of the servo.

5.The Method of Reducing the Occupancy of Color Discrimination(2)

5.1 Edge-Detecting

By searching information, we found out that one of the ways of edge detecting is useful in this case. The term edge refers to the set of the pixels where the surrounding grayscale baring greatly, which is also a fundamental symbol for graphs. If we consider edges as a

place where a certain amount of spots' brightness's change, the edge detection will be roughly be calculating and analyzing the change of brightness. So edge only exist in target background and area, and it is a important basis for partition graphs.

To simplify, we can analyze edge detection in one-dimensional space, in which example, a line of data of the brightness. For example, we can say there is an edge between the fourth and the fifth spot. Due to the position of edge, a symbol of position, is not sensitive for gray scale. So the use of edge decline the amount of data and remove most information that can be defined as useless, remaining the most important attribute of structure. This can help to simplify the method of color discrimination on botball.

5.2 The Method of Reducing the Occupancy of Color Discrimination

After understanding the principle of the edge detection, we write a short test program, found that using this method, the processor can grab a frame images, with edge bounded, distinguish between sections, to identify the color again, will greatly improve efficiency.

5.3 Examples in Field

Soble Operator is simpler and more widely used and have a better efficiency than Canny edge detection, but the edge detecting is not as accuracy as Canny's. However in many practical occasions, sobel edge is favored, especially when the efficiency is required, and the fine texture is less concentrated on. The operator of color discrimination given by the Botball fits this target.

Soble edge detection is often related with directional, it can detect the vertical edges or vertical or both. So we first define the two gradient direction coefficient:

kx=0;ky=1;% horizontal

kx=1;ky=0;% vertical

kx=1;ky=1;% both

Then we calculate the gradient image. We know that the edge point is the point where grayscale varies greatly, so the gradient image is calculated first and the lighter parts of the gradient image are extracted from it.

Sobel operator uses a 3 * 3 filter to filter the gradient image; we won't discibe the details of filtering or its meanings here.

Vertical filter: y_mask = op = [- 1-2-1; 0 0 0; 1 2 1] / 8;
Horizontal filter: op transposed: x_mask = op ';

After defining filter, we began to analyze the vertical and horizontal of the gradient image separately. This procedure can be done with the filter and dispose the image:

Bx = abs (filter2 (x_mask, a));
By = abs (filter2 (y_mask, a));

(Bx refers to the horizontal gradient of the image while By refers to the vertical gradient image)

6.Conclusion

As being an important program in the botball competition, color discrimination's appearance is always low for its high occupancy of resources. The teams which used this method also suffered from strict limitations. Both the two methods we offered can solve this problem effectively. Hope that these can give inspirations to other teams and can workout more effective solutions.

References :

1.A New Edge-extracting Technique Based on Soble Operater

http://wenku.baidu.com/link?url=9Pl36SdivGl32DK_LPbIX_Pp86Fw--QotdCUS-GKmlfOIeNhPsPW8QCCr4IjjGPS2kGMNSFwDGrsbC3pfuqUgLFu_a3Re7kTPqO19tTgpua

2.About Soble Operater

http://www.docin.com/p-1077517488.html

3.作者 40 年后谈 Sobel 算子的由来和定义

(A Discussion on Definition of Soble Operator and its Resource)

http://blog.sciencenet.cn/blog-425437-776050.html

4.彩色图像的边缘检测

(Edge Detecting for Colorful Image)

http://wenku.baidu.com/link?url=sihgRCjDGjAO5Jbp1jGwLAOWVjnxIvMELJsms49wGrTCgfJYOYJpOKttb41_uivxHwxEvVTsZLvuy-MbCpZ6dQH8Be1kfS9abvl-4VlKZ0W