Journal of Computers Vol. 30 No. 5, 2019, pp. 290-303
doi:10.3966/199115992019103005023

An Adaptive Robot Soccer Image Segmentation Based on HSI Color Space and Histogram Analysis

Yunxiu Xu, Bingxia Shen*, Mi Zhao, Shan Luo

School of Information and Communication Engineering,
Beijing Information Science & Technology University, Beijing, 100101, China
xuyunxiu@foxmail.com; shenbingxia@bistu.edu.cn; 18811536982@163.com; LuoShan0713@163.com

Received 30 March 2019; Revised 30 June 2019; Accepted 23 September 2019

Abstract. We proposed an adaptive image segmentation method based on HSI color space, which can be used for object identification in robot soccer games. This method requires no human intervention, and can guarantee the real-time performance and stability of segmentation to cope with the high real-time requirement and uneven illumination in the competition. Traditional image segmentation methods usually require high computer performance, which cannot be applied to the limited computational power of the robot. The results shows our method has highly accuracy and stability in most of situations.

Keywords: HSI color space, image histogram, image segmentation, robot soccer

1 Introduction

RoboCup (Robot World Cup) is an international initiative to foster the research in artificial intelligence (AI) and mobile robotics, by offering a publicly appealing, but formidable challenge. It is a perfect combination of sport and technology, thus has attracted many researchers and students. Since founded in 1997, it has promoted the research field for almost two decades [1-2].

In RoboCup MSL (Middle size league) teams of five fully autonomous robots play soccer with a regular size FIFA (Fédération Internationale de Football Association) soccer ball. Teams are free to design their own hardware but all sensors have to be on-board and there is a maximum size and weight for the robots. The research focus is on mechatronics design, control and multi-agent cooperation at plan and perception levels [3].

The MSL robot uses omni-directional vision system to self-locate and identify the situation in the field, the vision system is shown in Fig. 1. The left picture in Fig. 1 is the full view of the MSL robot; the upper right picture is the hardware part of the omni-directional vision system which including an upward industrial camera and a downward convex mirror; the lower right picture is the image which can be seen via vision system. The robot needs algorithm which can extract information in the image to judge the situation in the field and give the next strategy, like the positions of ball, field lines and the obstacles.

During robot competition and debugging, the light source, the camera's white balance and exposure settings, the interferences objects can be different, they all means our robot needs a well performance algorithm to adapt to different scenarios. And the speed of the MSL robot is fast, slow processing can cause problems with the robot.

MSL robot is a system with high real-time requirement. We aim at image segmentation to propose an adaptive image segmentation algorithm which enable the robot to segment obstacles, field, field lines and the ball in a high FPS, and can maintain stable performance under various abnormal white balance and exposure conditions.

* Corresponding Author

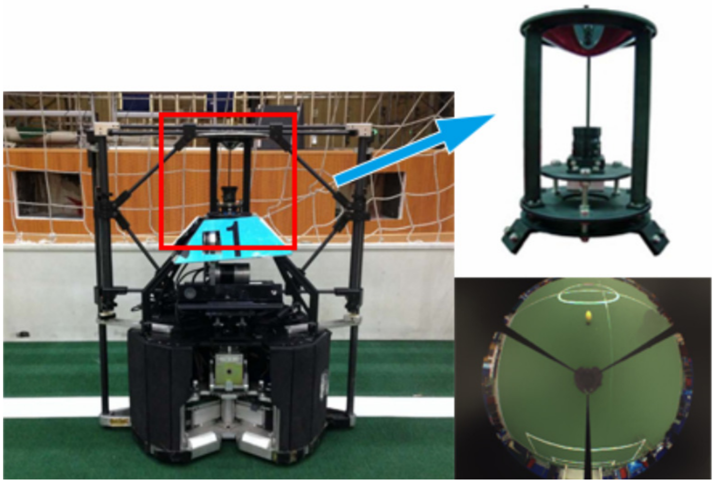


Fig. 1. Vision system

2 Related Works

With the deepening of robot soccer research, many methods and improvements have been proposed for the adaptive image segmentation and recognition of robot soccer in the early stage: Liu et al. [4] proposed an improved CLUT (Color Look Up Table) color classification method for robot soccer. This method was efficient and convenient to establish look up table and to segment similar colors in RoboCup. Fitriana et al. [5] propose a system that uses the combination of color based segmentation and feature detection to detect the color and also the shape feature of the object used in the soccer robot game. Qian and Lee [6] propose an efficient approach for adaptive soccer field detection model utilizing NAO's two-camera system. Härtl et al. [7] propose a new object recognition system where objects are found based on color similarities. Moreover, white balance and camera gain are also effective methods. Bailey et al. [8] describe an algorithm for automatic gain control and white balancing within the camera, this can significantly reduce the effects of the variations in lighting on the thresholds. At the same time, the convolutional neural network is used to identify robot football: Speck et al. [9] presents a neural approach using a convolutional neural network to localize the ball in various scenes. However, machine learning algorithms needs to provide plenty of training datasets, it is a time consuming work for over robot to collect many kinds of datasets. Although the above works have solved the problem of image segmentation of soccer robot, they cannot adapt to the situations of various light and field conditions without artificial.

3 Proposed Method

We divide the image segmentation algorithm into two parts. The first part is the pre-processing, which can accurately separate various objects in the image. However, due to the large amount of computation, it cannot adapt to the real-time requirements. The illumination condition will not change much in a short time, so we need to accelerate the image segmentation algorithm. We save the results of preprocessing in the form of HSI range values, and start to segment the images directly through the saved threshold. The processing speed of this process is very fast and can meet the real-time requirements of the algorithm. The algorithm flow chart is shown in Fig. 2, and the image to be extracted is shown in Fig. 3.

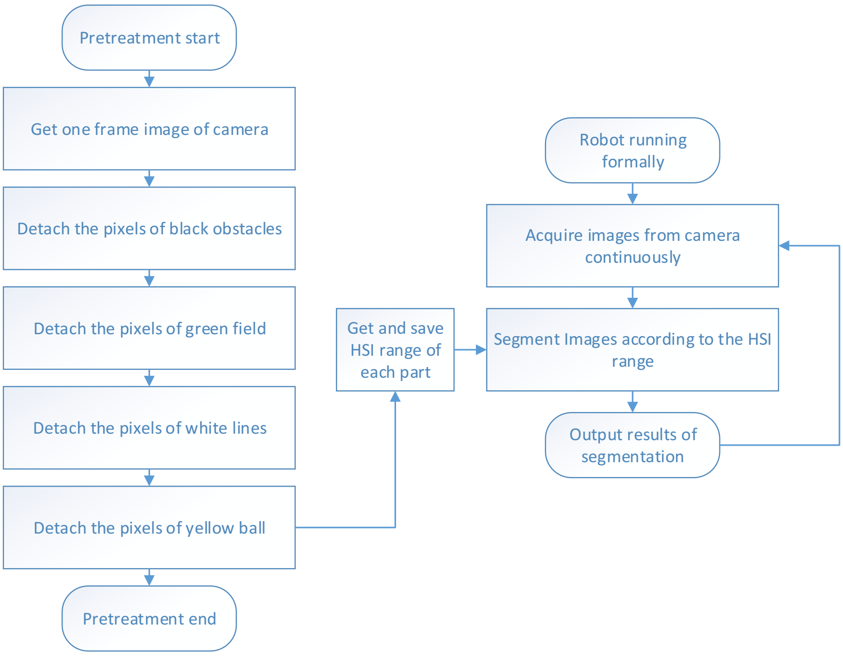


Fig. 2. System flowchart

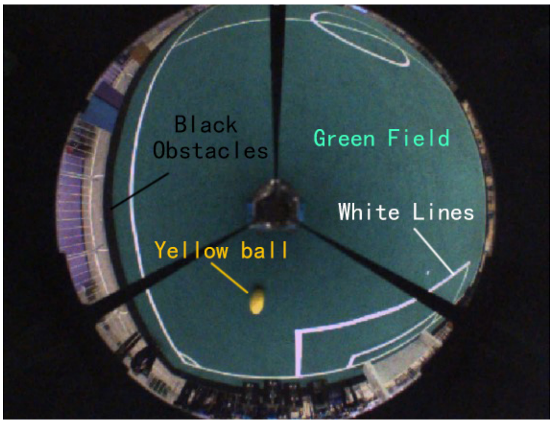


Fig. 3. The part to extract

3.1 Black Obstacle Extraction Algorithm

In the rules of RoboCup MSL, The base color of a robot's body must be black. every team is expected to try hard to hide non-black parts of the robot as much as possible, especially parts that have colors used for the ball or the field of play [10]. Therefore, the black parts in the picture should be regarded as obstacles, the pixel in the black obstacle part of the image is characterized by lower brightness than the pixel in other parts. Therefore, finding a threshold that can separate the black obstacles from the other parts can completes the segmentation. Firstly, the intensity histogram of the image is analyzed and shown in Fig. 4.

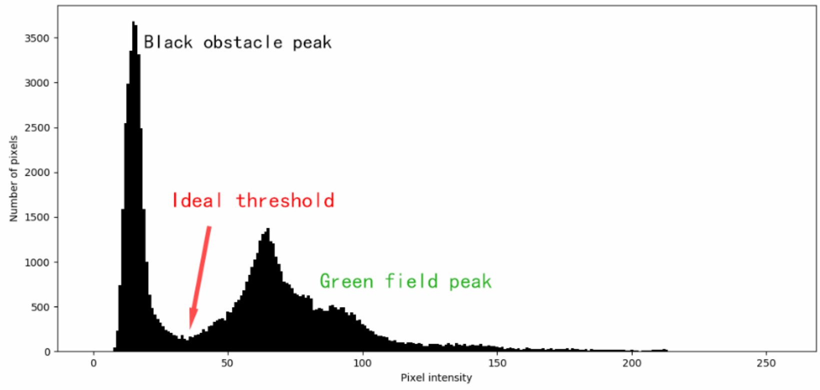


Fig. 4. Intensity histogram of the image

According to Fig. 4, the areas with the largest proportion are black obstacles and green fields.

They appear as two peaks in the histogram of Fig. 4. And the bottom between the two peaks is easy to find, that's the ideal threshold. Due to the fact that there will be no more extreme illumination during the competition, the actual performance of this algorithm is relatively good and is shown in Fig. 5.

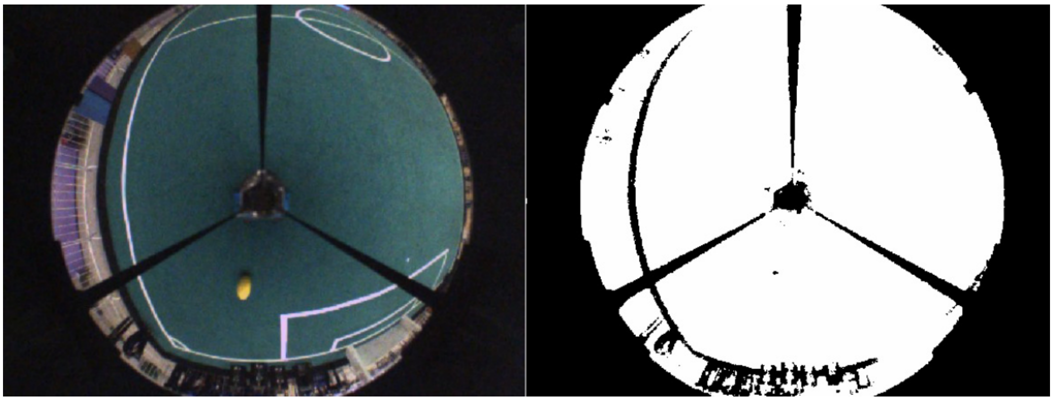


Fig. 5. Black obstacle extraction results

3.2 Green Field Extraction Algorithm

Step 1. Filter out the noise. In terms of brightness, the difference between the green field and the non-green area such as the ball, goal and the outside of the field is not very large, but its most obvious feature is its constant hue. However, due to the large amount of noise generated by the camera, the hue of the noise is quite different from the green field, which will seriously interfere with the identification results. So the filter algorithm should be used to remove the noise in the image.

In image segmentation, Preserving the clarity of the boundaries of all parts is of paramount importance, and the Edge Preserving Filter refers to a kind of special Filter that can effectively preserve the Edge information of images in the filtering process, which is very suitable for this kind of scene. Surface Blur is a kind of edge preserving filter [11], Its implementation is relatively simple, Surface Blur algorithm is used to filter the image, as shown in equation 1:

$$X_{out} = \frac{\sum_{i=1}^{(2r+1)^2} [(1-\frac{|x_i - x|}{2.5Y} x_i]}{\sum_{i=1}^{(2r+1)^2} (1-\frac{|x_i - x|}{2.5Y} x_i)}. \quad (1)$$

Where, r represents the convolution kernel radius, Y represents the threshold value, x represents the current pixel value to be processed, x_i represents the ith pixel value in the convolution kernel, and x_{out} represents the output result.

The convolution kernel is used for convolution operation, and the filtering effect with clear edges can be obtained. The noise points can be effectively filtered, and the edges can be clearer, so that the algorithm can correctly segment the image.

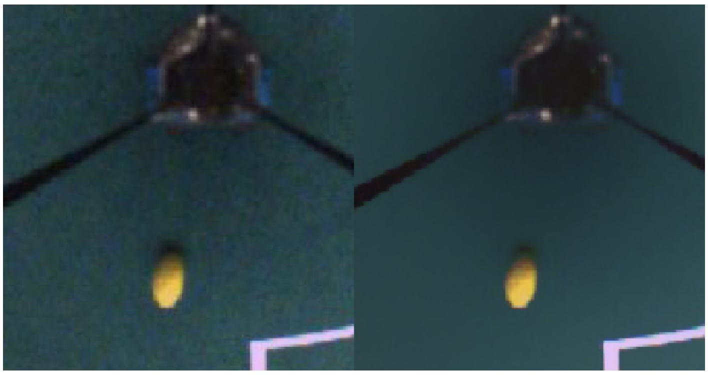


Fig. 6. The result of the edge preserving filter

Step 2. Obtaining and smoothing of hue histogram. In HSI color space, The hue value of the green field is more prominent, Therefore, we conduct histogram analysis of filtered image. Firstly, the image is transformed into HSI color space and the hue values of all pixels are extracted and the histogram of the hue values is drawn as *GreenHueHist*.

Due to the largest number of green pixels, there will be an obvious peak in the histogram. As long as the boundary value of this peak can be extracted, the green field can be separated. But the boundary value of this peak is fuzzy and difficult to determine. So the histogram smoothing method is used to smooth the histogram. Interpolation based histogram smoothing method can quickly combine the histogram peaks and the surrounding complex signals into wave peaks with obvious single boundary. The interpolation - based histogram smoothing method is calculated by equation 2.

$$temp = \frac{1}{step} \sum_{i=-\frac{step}{2}}^{\frac{step}{2}} f(x+i). \tag{2}$$

Step is the step size, which is positively correlated with smoothness, and then get the smoothed green hue histogram *GreenHueSmooth.*

Step 3. Histogram peak boundary acquisition. After obtaining the smooth hue histogram, the peaks of the more obvious green parts are obtained. The first order differential of *GreenHueSmooth* is carried out to obtain the histogram change rate function. The two highest points of this function are the boundary of the peak.

Take the boundary of two peaks as the range value of hue, filter all pixels, the segmentation image of the green field can be obtained. The results of histogram smoothing and derivative peak are shown in Fig. 7 and Fig. 8, and the ultimate extraction results are shown in Fig. 9.

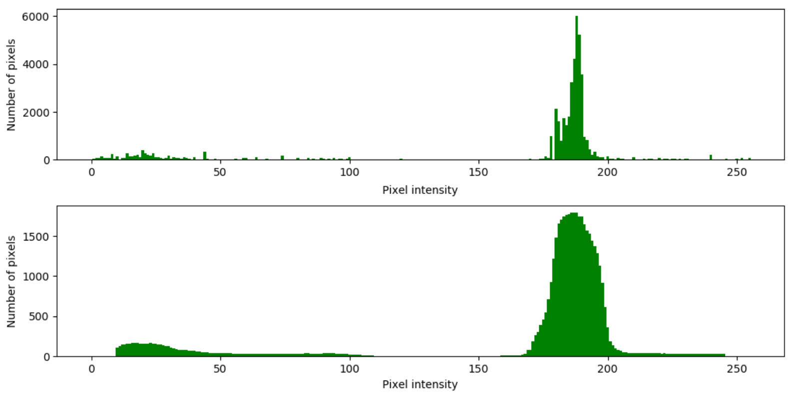


Fig. 7. Hue histogram of filtering results (top) and histogram smoothing results (bottom)

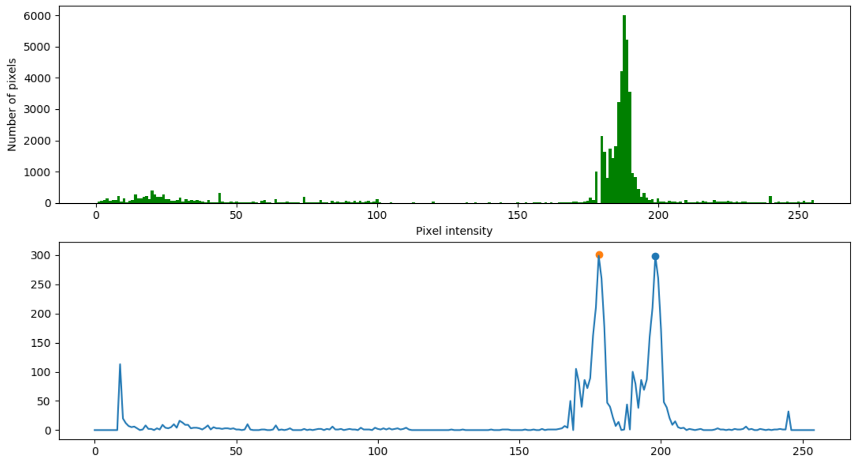


Fig. 8. The peak of derivative matches the boundary of the peak of hue

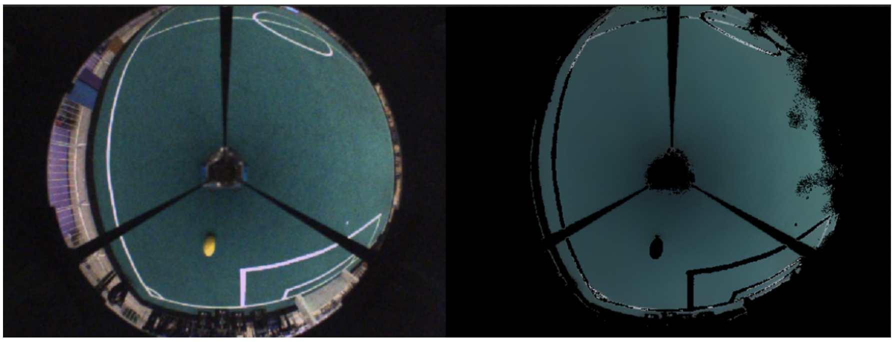


Fig. 9. Green field extraction results

3.3 White Field line Extraction Algorithm

The accuracy of robot self-positioning depends most on the white field line, so the recognition effect of the white field line is very important. Similar to the black obstacles, white field lines have extreme values in intensity. In the intensity histogram, the pixels of the white field line creates a higher peak on the right side of the histogram. This peak concentrates most of the pixels of the white field lines. Most white field lines can be extracted from this way, as shown in Fig. 10.

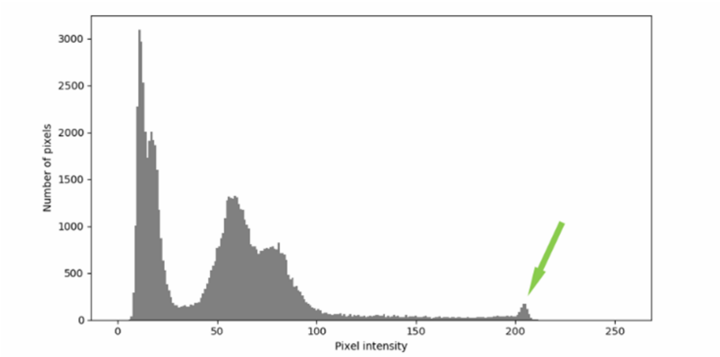


Fig. 10. Distribution of white field lines in the intensity histogram

However, due to the dispersion of the lens, the setting of white balance and other issues, the brightness and color of the white field line pixel are not uniform, and other colors different from white also appear, as shown in Fig. 11. But what these white field lines have in common is that they have higher intensity and lower saturation. So we can separate it from the rest based on the properties.

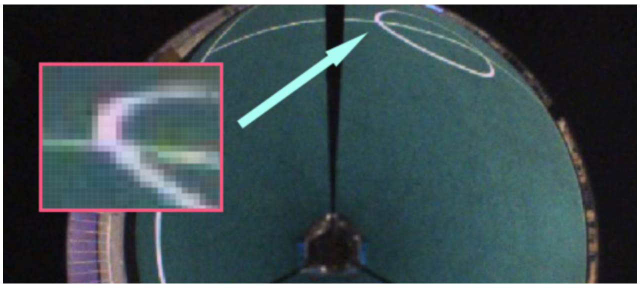


Fig. 11. The white part of the image edge will show some degree of partial color

On the left side of the white line peak on the right in Fig. 10, there is a long section of relatively flat part, where the field line with low brightness at the edge of the lens and other pixels with high brightness, such as the ball and the interference outside the field are inside. Therefore, the following steps are proposed to extract white field lines:

Step 1. The peaks distributed to the right of intensity histogram and the pixels in the flat area to the left are extracted, as shown in Fig. 12.

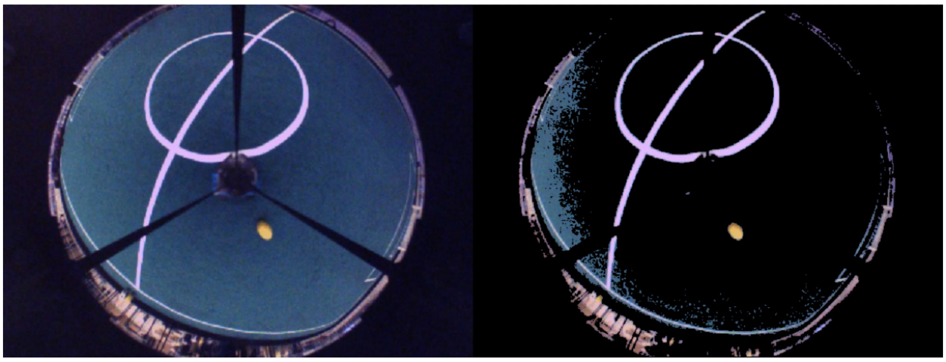


Fig. 12. The brightness alone will extract the white line and there will be more interference (result of the first step)

Step 2. Get the saturation of the white field line from the pixel in the right peak.

Step 3. Filter the pixels extracted in the first step again with the saturation obtained in the step 2, as shown in Fig. 13.

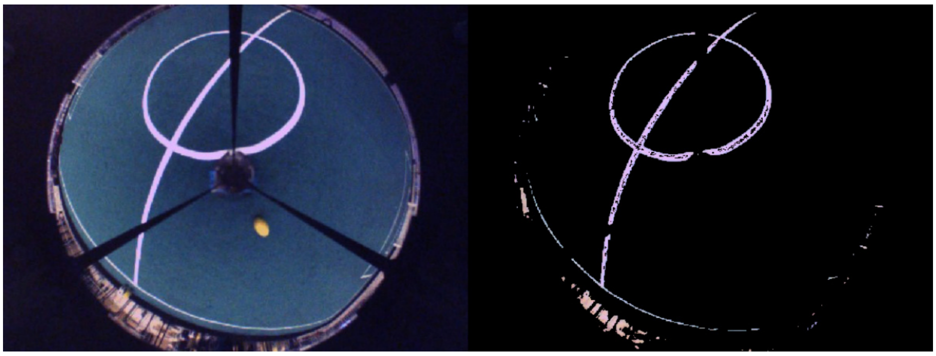


Fig. 13. Field lines are extracted using both brightness and saturation (result of step 3)

3.4 Yellow Ball Extraction Algorithm

In the MSL RoboCup competition, a full-size soccer ball is required, but the soccer ball is painted yellow [10]. In the competition field, the brightness and saturation of the yellow ball are high, and the ball can be separated according to these two characteristics. It can be seen from Fig. 12 and Fig. 14. In the first step of white line extraction, the yellow ball is also extracted. Because the ball has a higher brightness, it can continue to extract the ball based on the high saturation of the ball. The following methods are proposed:

Step 1. It is processed in the result of the first step of white line extraction, converted to HSI color space, filtered with saturation range [*a*, *Max*], and reserve only the pixels in this range. ‘*a*’ began to decrease gradually from *Max* and used morphological filtering to filter out noise points, as shown in Fig. 16. The initial saturation range is narrow, so few pixels are filtered out and the result is all black. Continue to reduce the ‘*a*’ until the first valid block (number of pixels greater than a certain value) appears in the result. Record the center coordinates of this pixel block, as shown in Fig. 14.
Step 2. Continue the process of the first step, and use the contour detection algorithm to find the center of the pixel block with the largest area in each obtained result. When the center is not included in the pixel block of the first step, the iteration stops. The result is shown on the left of Fig. 15.
Step 3. When the iteration stops, only the pixel block where the center coordinate of the pixel block recorded in the first step is kept, and the pixel where the ball is can be obtained by clearing all other parts of the image. The result is shown on the right of Fig. 15.

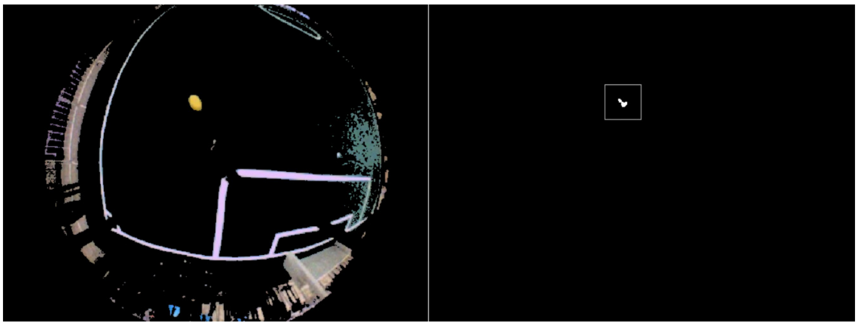


Fig. 14. Before filtering (left), the first time the state of a valid pixel block appears after filtering (right, indicated by box)

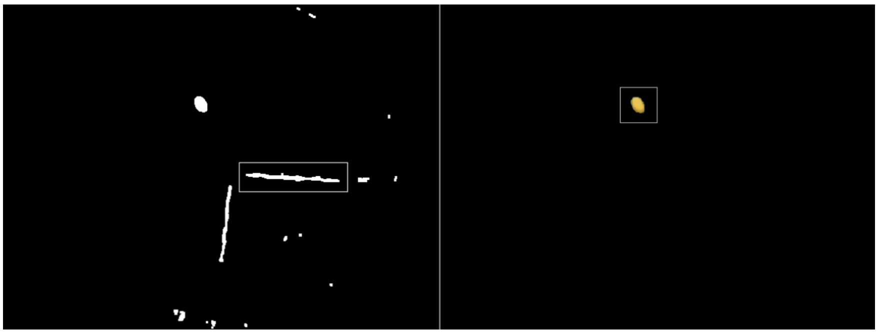


Fig. 15. Results after the iteration stops (left) and final results (right)

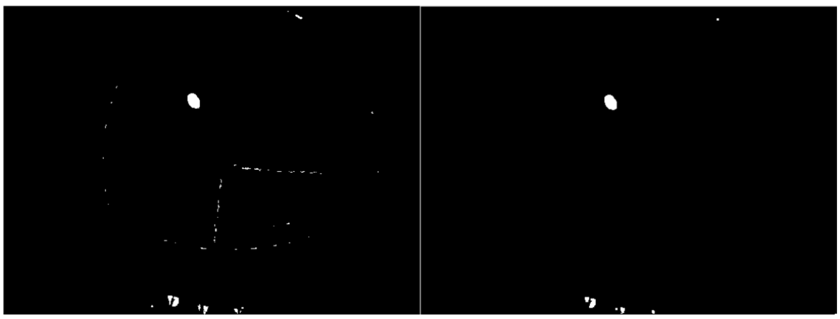


Fig. 16. Morphological filtering on the image (before in the left and after in the right)

3.5 Acceleration of Image Segmentation Algorithm

Through the process above, the function of automatic image segmentation is realized. However, the processing time of the above algorithm is relatively long. In the competition, the robot moves at a fast speed and its limit speed is close to 5m/s, so the speed of image processing is extremely high, and the processing speed should reach more than 30fps. The processing speed of the above algorithm is lower than 1Fps, so a faster image segmentation method should be used on the basis of the above algorithm.

Step 1. The images of the four parts separated by the above algorithm are converted into HSI color space, and all pixels are traversed to obtain the upper and lower limits of H, S and I values respectively.

Step 2. Modify some of the values in the result obtained in the first step. The hue of the black obstacles and the white field line are meaningless, so regardless of the results obtained in the first step, they are modified to the lowest and highest values (usually [0, 255]). The white field line is usually the brightest part of the image, so change the upper limit of its brightness value to the highest (usually 255). Save all threshold results.

Step 3. For later camera input images, there is no need to go through the first and second steps of the process. Traverse the whole image, and then filter according to the ranges saved in the second part to get the segmentation result of the four parts.

After running the previous work of the first step and the second step, the subsequent image segmentation only needs to continue to execute the process of the third step and only need to traverse the image once to complete the image segmentation. We represent the segmented results in different colors for easy observation. Black obstacles are represented by purple, green fields are represented by green, white field lines are represented by blue, and yellow balls are represented by red. The visualization is shown in Fig. 17.

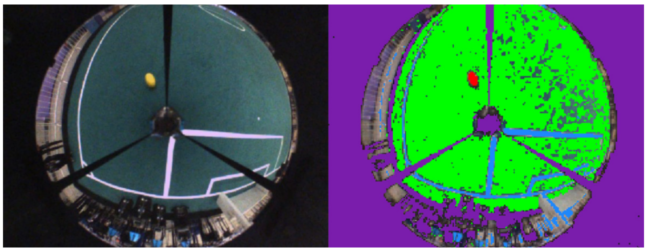


Fig. 17. Results of fast image segmentation method

4 Experimental Results

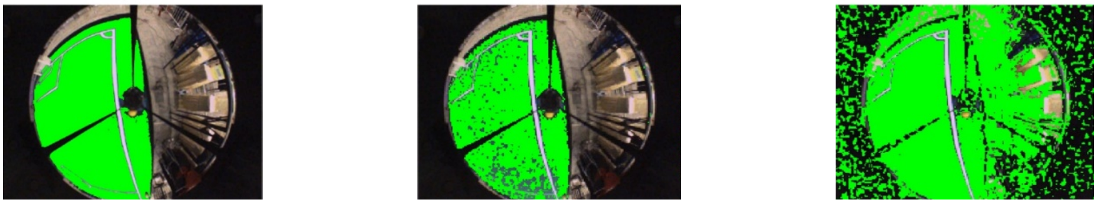
In order to evaluate our proposed algorithm, we use two evaluation methods to determine the results of image segmentation. First method is to determine whether the robot can correctly complete the movements of moving, avoiding obstacles and catching balls. The second method is to compare the pixels in the results of each part with ground-truth to get the accuracy. The judging criteria are shown in equation (3).

$$Accuracy = \frac{PixT - PixF}{PixG} \times 100\% \tag{3}$$

Where, $PixT$ is the correct number of pixels calibrated in the results, $PixF$ is the wrong number of pixels, and $PixG$ is all the pixels in ground-truth.

4.1 Effect of Edge Preserving Filter on Green Field

The output picture of the camera has a lot of noise, and the hue of these noises is quite different from green fields. Therefore, if no filtering is carried out, the hue range of the calculated results will be much larger than the actual value, resulting in a large recognition range. The final recognition effect can be seen from Fig. 18. The use of edge preserving filter can obviously keep the segmentation result within a correct range. In Fig. 18(b), the Hue range of the green site was 165-212, while in Fig. 18(c), the Hue range was 15-227, covering 60% of the Hue range, obviously too large.



(a) Ground-Truth

(b) Identification results of green field after using edge preserving filter, Hue range is 165-212

(c) The result of identifying green field without using filter, Hue range 15-227

Fig. 18.

3.2 Circumstance of Normal Exposure, Normal Color Temperature and Less Interference

When the color temperature and the exposure is normal, and there is less off-site interference in the image, the final recognition results of each part are shown in Fig. 19. Compared with ground-truth, the statistical results show that the segmentation accuracy of black obstacle is 97.3%, that of green field is 91.1%, that of white field line is 73.5%, and that of yellow ball is 89.3%. Under this condition, the robot can work normally, which proves that the segmentation result is correct.

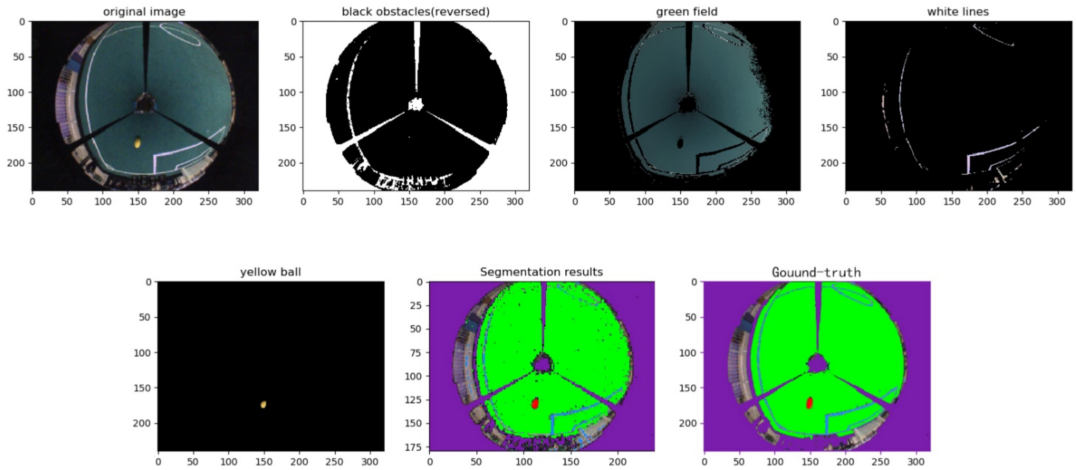


Fig. 19. The segmentation result of a common situation

4.3 The Situation of Color Temperature Exposure Has a Large Deviation and a Lot of Interference

4.3.1 A lot of Interference

Due to the very large visual range of omni-directional vision sensor of robot and the limitation of competition field, there may be a lot of external interference. Fig. 20(a), Fig. 20(b) and Fig. 20(c) respectively show three kinds of interference of objects in the field and interference of other colors outside the field. It can be seen that the image can be basically correctly segmented in both cases, mainly because the interference does not change the fact that the green and black areas are relatively large and the white field lines are the brightest.

4.3.2 Change of Color Temperature

Under the lighting of different light sources, the color temperature of the picture usually has a great difference, which may have a great impact on the segmentation. The biggest influence of the change of color temperature is the identification of the ball and the green field. After the color temperature changes, the color of the object in the picture is heavily distortion, but still can be easily separated from the hue value, Therefore, the image segmentation under different color temperature conditions (+-40%) can still maintain greater accuracy. The segmentation results are shown in Fig. 20 (d) and Fig. 20 (e).

4.3.3 Change of Exposure Value

Different brightness of light source has great influence on histogram distribution of image. During the operation of the robot, the brightness of the light source often changes, especially in sunlight. Therefore, the algorithm was tested in the case of overexposure and underexposure (+ -1.4ev). Test results are shown in Fig. 20(f) and Fig. 20(g). In the case of overexposure, the brightness value of many pixels overflows, resulting in a large number of brighter pixels in the picture. However, the overall distribution of the histogram has not changed, so it has little impact on the recognition. In the case of underexposure, black obstacle, green field and yellow ball recognition can be more normal, but the pixels of white lines are few, by histogram analysis, as shown in Fig. 21, we can learn that most of the pixels are on the darker area, and black and green field two peak still exists, so they can still be identified. However, the small peak on the right of the white field line has completely disappeared, and the flat area on the left has become a ladder. The algorithm only recognizes a few pixels in the first step. This is a relatively rare condition that requires prior exposure correction.

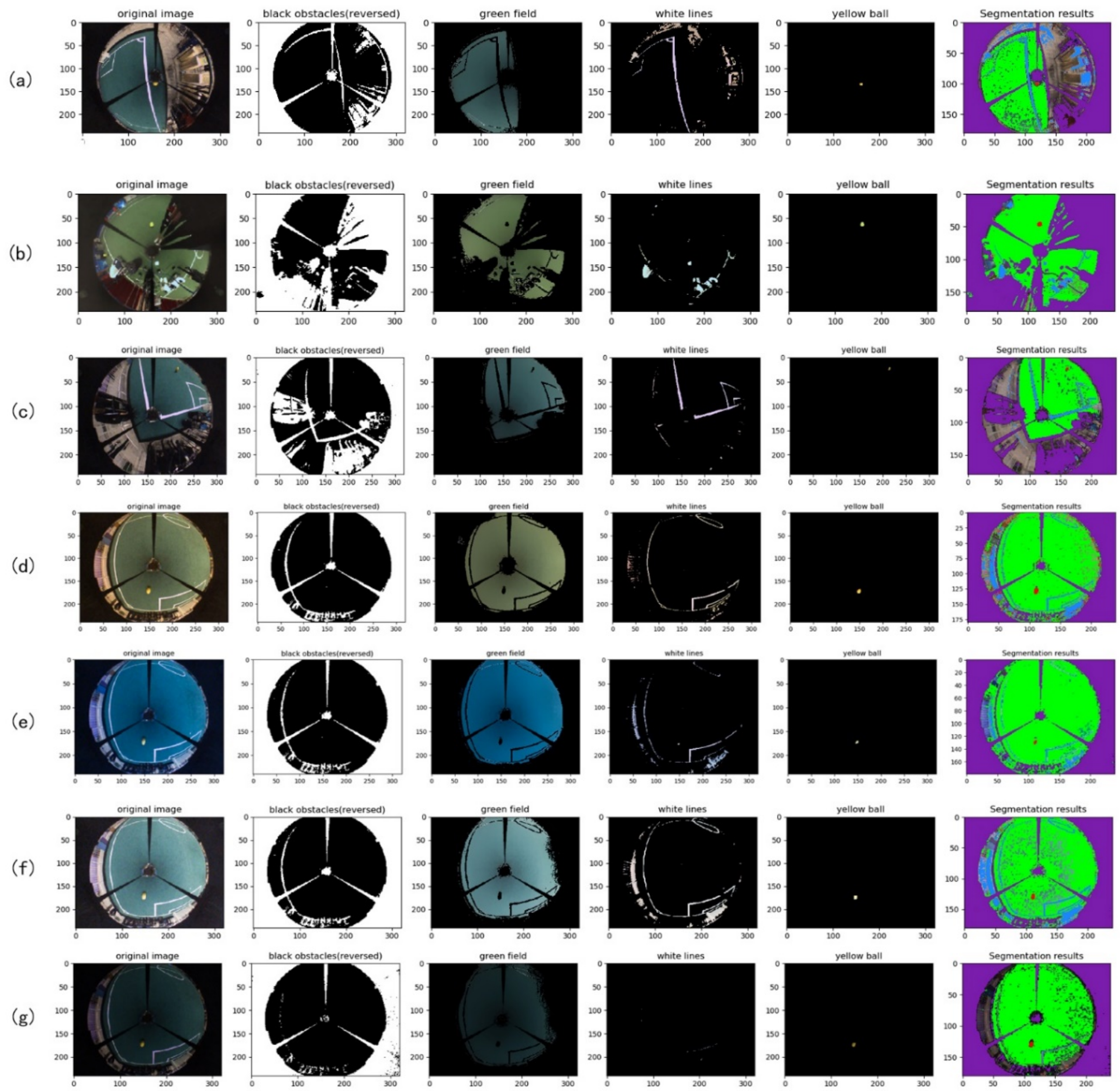


Fig. 20. Result of multiple interference situations

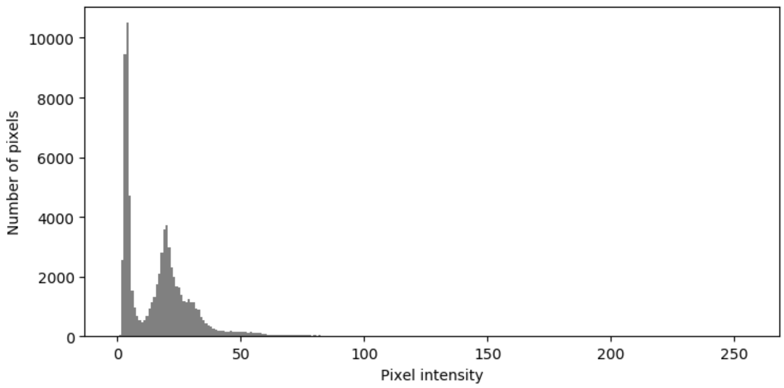


Fig. 21. Intensity histogram of under-exposure situation

The segmentation results in Fig. 20 are used as the standard for normalization, and the accuracy of each parameter is regarded as 100%. Meanwhile, the subsequent algorithm will ignore the pixels of the white field line outside the green field, so the pixels of this part of the white field are not included in the statistics, and are summarized in Table 1. At the same time, change the camera parameters or robot position so that the picture is the same as above, then run the algorithm. The segmentation results are applied to the robot. In the case of (a) to (f), the robot can complete the actions of moving, obstacles avoiding and ball catching. In (g) situation, the robot is unable to conduct self-positioning and ball positioning due to abnormal field line segmentation, and unable to complete the action. According to the actual experiment of the robot, it can work well when the segmentation accuracy is above 70%. Therefore, we do not need to get the optimal value, but to ensure that the accuracy is always stable at more than 70% and can adapt to various environmental changes.

Table 1. The accuracy of the results in different conditions

Condition	Black obstacle	Green field	White line	Yellow ball
(a)	93.3%	97.4%	95.5%	61.0%
(b)	105.1%	109.5%	97.7%	96.6%
(c)	95.6%	97.3%	89.5%	73.5%
(d)	93.1%	78.9%	85.0%	81.1%
(e)	98.6%	135.7%	58.6%	97.3%

I7-6770hq CPU was used as the test environment. The test results of each part of the algorithm are shown in Table 2. The total time of image segmentation is 1.4ms. And the highest frame rate of the camera is 120Fps, so the real-time performance of the segmentation algorithm is very well.

Table 2. The elapsed time for different steps under specific hardware conditions

Condition	Intel (R) Core (TM) i7-6770HQ CPU 2.6GHz
Distortion correction	1.2ms
Image binarization	2.1ms
Image segmentation	**1.4ms**
Field line matching	1.1ms
self-localization	0.6ms
Path planning	1.4ms
Speed fitting	0.4ms
Total time consuming	8.2ms

The following is the experiment of applying our algorithm to other non-MSL robot's situation, as shown in Fig. 22. The algorithm can complete the image segmentation task in some similar situations (for example, background with large areas of similar colors, markers with high brightness such as white lines, and shadows or obstacles with dark colors). The algorithm can be applied to more image segmentation scenes after subsequent adaptation and improvement.

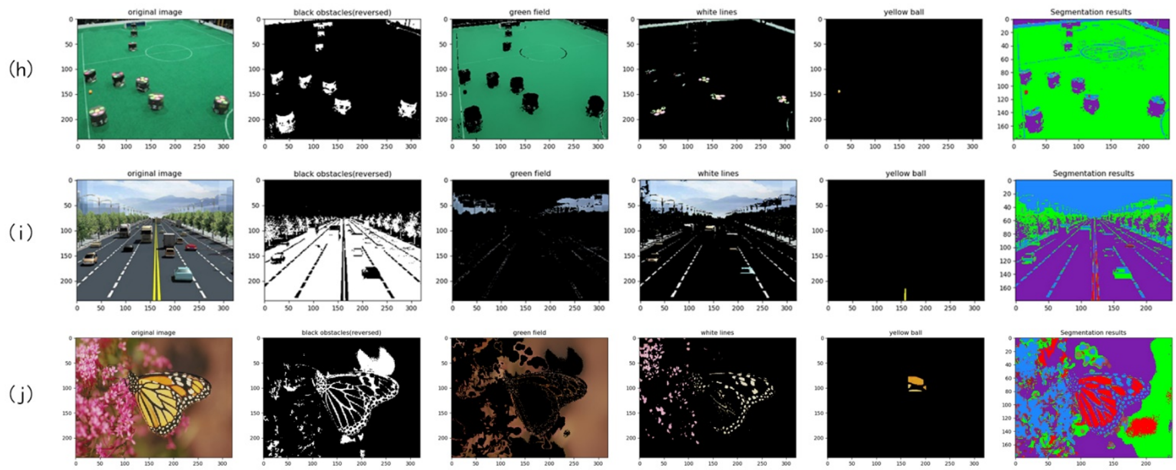


Fig. 22. Results of some non-MSL robot scenarios

5 Conclusion

We propose an adaptive image segmentation method based on HSI color space for MSL soccer robots. By operating the characteristics of histogram of images in different color Spaces, we complete the extraction of pixels in different areas of an image. This method does not need human intervention and can automatically complete the segmentation of different areas in the picture. After obtaining the HSI value range once, the image segmentation can be completed only by comparing the subsequent images with the HSI range. At the same time, we tested the algorithm under different extreme color temperature, exposure, object interference and other situations, and the results showed that the algorithm could meet the requirements in most situations with a certain robustness. More importantly, we explore the segmentation ability of this method by using some scenes with similar characteristics to MSL soccer fields, which can also effectively segment these images. In the future, we will explore more extensive image segmentation methods based on this algorithm. This algorithm greatly reduces the amount of work needed to calibrate the colors under the condition of changing the field or natural light, and also enables the robot to complete the game more easily.

Acknowledgements

This work is supported by Talent Training Quality Improvement Foundation of Beijing Information Science & Technology University (No. 5101924100)

References

[1] H. Kitano, M. Asada, Y. Kuniyoshi, I. Noda, E. Osawa, Robocup: the robot world cup initiative, in: Proc. the First International Conference on Autonomous Agents, 1997.

[2] H. Kitano, M. Asada, Y. Kuniyoshi, I. Noda, E. Osawa, H. Matsubara. Robocup: a challenge problem for ai, AI Magazine 18(1)(1997) 73-85.

[3] RoboCup, HOME OF THE MSL. <https://msl.robocup.org>.

[4] F. Liu, H. Lu, Z. Zheng, A modified color look-up table segmentation method for robot soccer, in: Proc. 4th Latin Amercia Robotic Symposium/IX Congreso Mexicano de Robotica 2007(4th) IEEE LARS/COMRob07, 2007.

[5] A.N. Fitriana, K. Mutijarsa, W. Adiprawita, color-based segmentation and feature detection for ball and goal post on mobile soccer robot game field, in: Proc. 2016 International Conference on Information Technology Systems and Innovation (ICITSI), 2016.

[6] Y. Qian, D.D. Lee, Adaptive field detection and localization in robot soccer, in: Proc. 20th RoboCup International Symposium, 2016.

[7] A. Härtl, U. Visser, T. Röfer, Robust and efficient object recognition for a humanoid soccer robot, in: Proc. Robot Soccer World Cup, 2013.

[8] D. Bailey, M. Contreras, G.S. Gupta, Towards automatic colour segmentation for robot soccer, in: Proc. 2015 6th International Conference on Automation, Robotics and Applications (ICARA), 2015.

[9] D. Speck, P. Barros, C. Weber, S. Wermter, Ball localization for robocup soccer using convolutional neural networks, in: Proc. Robot World Cup, 2016.

[10]RoboCup, MSL Technical Committee, Middle size robot league rules and regulations for 2019, Version-20.020181231, 2018.

[11] F. Banterle, M. Corsini, P. Cignoni, R. Scopigno, A low-memory, straightforward and fast bilateral filter through subsampling in spatial domain, Computer Graphics Forum 31(1)(2012) 19-32.