



A robust omnidirectional vision sensor for soccer robots

Huimin Lu ^{*}, Shaowu Yang, Hui Zhang, Zhiqiang Zheng

Department of Automatic Control, College of Mechatronics Engineering and Automation, National University of Defense Technology, Changsha, China

ARTICLE INFO

Keywords:
Omnidirectional vision
Soccer robots
Camera parameters auto-adjustment
Ball recognition
RoboCup

ABSTRACT

How to make robot vision work robustly under varying lighting conditions and without the constraint of the current color-coded environment are two of the most challenging issues in the RoboCup community. In this paper, we present a robust omnidirectional vision sensor to deal with these issues for the RoboCup Middle Size League soccer robots, in which two novel algorithms are applied. The first one is a camera parameters auto-adjusting algorithm based on image entropy. The relationship between image entropy and camera parameters is verified by experiments, and camera parameters are optimized by maximizing image entropy to adapt the output of the omnidirectional vision to the varying illumination. The second one is a ball recognition method based on the omnidirectional vision without color classification. The conclusion is derived that the ball on the field can be imaged to be an ellipse approximately in our omnidirectional vision, and the arbitrary FIFA ball can be recognized by detecting the ellipse imaged by the ball. The experimental results show that a robust omnidirectional vision sensor can be realized by using the two algorithms mentioned above.

© 2010 Elsevier Ltd. All rights reserved.

1. Introduction

The RoboCup Middle Size League (MSL) competition is a standard real-world test bed for autonomous multi-robot control, robot vision and other relative research subjects. In the past decade, omnidirectional vision system has become one of the most important sensors for the RoboCup MSL soccer robots, for it can provide a 360° view of the robot's surrounding environment in a single image. Robots can use it to realize object recognition [1], tracking [2] and self-localization [3,4] which provides perception information about the environment for robot control, planning, multi-robot cooperation and coordination.

The final goal of RoboCup is that the robot soccer team defeats human championship team, so robots will have to play competitions under highly dynamic even outdoor lighting conditions, and without the constraint of the current color-coded environment. According to the current rule of RoboCup MSL, the illumination is not specified, color goals have been replaced with white goal nets, color goalposts have been removed, and the technical challenge of playing with an arbitrary FIFA ball has been introduced. So there are two challenging research focuses in robot vision for soccer robots, although they have been researched for many years. First, it is still not easy to make vision system work robustly under varying lighting conditions in the color-coded RoboCup MSL environment [5]. The traditional methods, such as segmenting the image first

and then detecting the color blobs by using a color lookup table calibrated off-line, would not work well when the lighting condition fluctuates during the competition. Second, it is much more difficult to recognize generic objects such as ordinary FIFA balls than to recognize the color-coded objects such as orange balls. So it would be significant to develop robust omnidirectional vision sensors that can handle these issues for the improvement of the soccer robots' performance and the realization of the final goal of RoboCup. In this paper, the term "robustness" means not only that robot vision can work well and consistently under varying lighting conditions, but also that generic objects, like the FIFA balls with different colors and textures, can be well recognized.

Generally, the robustness of robot vision can be realized in the phase of image acquisition or in the phase of image processing [6]. The former one, which is often ignored by researchers, is to acquire images to describe the environment as consistently as possible under different lighting conditions by auto-adjusting camera parameters [1,7–9]. The camera parameters displayed here are image acquisition parameters, not the intrinsic or extrinsic parameters in camera calibration. The phase of image processing consists of many different approaches. For example, some researchers processed and transformed the images to achieve some kind of constancy, such as color constancy [10] by Retinex algorithm [11,12], and some others designed adaptive or robust object recognition algorithms [13,14]. So the images can be analyzed and understood robustly to a certain degree. In this paper, we also try to improve the robustness of our omnidirectional vision in image acquisition and image processing to deal with the two challenging issues for soccer robots. First, we propose a novel

* Corresponding author. Tel.: +86 731 84576455.

E-mail addresses: lhmmew@nudt.edu.cn (H. Lu), yangshaowu@hotmail.com (S. Yang), zhanghui_nudt@126.com (H. Zhang), zqzheng@nudt.edu.cn (Z. Zheng).

technique to auto-adjust the camera parameters based on image entropy to adapt the output of the vision system to varying lighting conditions in the image acquisition. Second, we design a robust FIFA ball recognition method based on the omnidirectional vision without color classification in the image processing.

In the following parts, related research will be discussed in Section 2. Then our omnidirectional vision system and its calibration method will be introduced briefly in Section 3. After that, in Section 4, we will propose the camera parameters auto-adjusting method based on image entropy. In Section 5, we will present the ball recognition method based on the omnidirectional vision without color classification. The experimental results and discussions will be included in the corresponding sections. Finally the conclusion and future work will be presented in Section 6.

2. Related research

The RoboCup MSL environment is currently color-coded, so it is a research focus to design robust vision systems to recognize color-coded objects in the RoboCup community. Besides adaptive color segmentation methods [13], color online-learning algorithms [15,16], and object recognition methods independent on color information [17,18], several researchers have tried to improve the robustness of vision sensors by adjusting camera parameters in the image acquisition. Grillo et al. defined camera parameters adjustment as an optimization problem, and used the genetic meta-heuristic algorithm to solve it by minimizing the distances between the color values of image regions selected manually and the theoretic values in color space [7]. The theoretic color values were used as referenced values, so the effect from illumination could be eliminated, but special image regions needed to be selected manually by users as required by this method. Takahashi et al. used a set of PID controllers to modify the camera parameters like gain, iris, and two white-balance channels according to the changes of a white referenced color which is always visible in the omnidirectional vision system [1]. Lunenburg and Ven adjusted the shutter time by designing a PI controller to modify the color of the referenced green field to be the desired values [8]. Neves et al. proposed an algorithm for autonomous setup of the camera parameters such as exposure, gain, white-balance and brightness for their omnidirectional vision [9], according to the intensity histogram of the images, a black and a white region known in advance. A color patch including the black and white region is required on the field, so it can only be used off-line before the competition. In these four methods, some referenced color is needed, so it is difficult to apply them directly in other situations.

Some similar research took place in digital still cameras and consumer video cameras. Many parameter adjusting mechanisms have been developed to achieve good imaging results, such as auto exposure by changing the iris or the shutter time [19], auto white-balance [20], and auto focus [21]. In some special multiple slope response cameras, the response curve can be adjusted to adapt the dynamic response range to different lighting conditions by automatic exposure control [22]. But all these methods are on the level of the camera hardware, so they are not applicable to most cameras used in robot vision systems except some special hardware-support cameras.

To realize the final goal of RoboCup, the current color-coded environment of RoboCup MSL should be further changed. One example would be replacing orange balls with generic FIFA balls. So it is significant to develop image processing algorithms to recognize arbitrary FIFA balls without color classification like human beings. According to our definition of “robustness”, the robustness of vision systems will also be improved greatly by these algo-

rithms, because no special color or texture information is needed in the object recognition.

A so-called Contracting Curve Density (CCD) algorithm [17,23,24] was proposed by Hanek et al. to recognize soccer balls without color labeling. This algorithm fits parametric curve models with image data by using local criteria based on local image statistics to separate adjacent regions. The contour of the ball could be extracted even in cluttered environments under different illumination, but the vague position of the ball needed to be known in advance. Therefore global detection could not be realized using this method. Treptow and Zell integrated the Adaboost feature learning algorithm into a condensation tracking framework [18], so a ball without a special color could be detected and tracked in real-time even in cluttered environments. Miti et al. presented a novel scheme [25] for fast color invariant ball detection, in which the edged filtered images serve as the input of an Adaboost learning procedure that constructs a cascade of classification and regression trees. Different soccer balls could be detected by this method in different environments, but the false positive rate was high when other round objects were introduced into the environment. Then they combined a biologically inspired attention system-VOCUS [26] with the cascade of classifiers. This combination made their ball recognition highly robust and eliminated the false detection effectively. Coath and Musumeci proposed an edge-based arc fitting algorithm [27] to detect the ball for soccer robots. Bonarini et al. used circular Hough transform on the edges extracted from a color invariant transformation algorithm to detect the generic ball, and Kalman Filter was also applied to track and predict the position of the ball in the next image to reduce the computational load [28]. However, all the algorithms mentioned above were used only in the perspective camera in which the field of view was far smaller and the image was also much less complex than that of the omnidirectional vision system.

Some researchers have used omnidirectional vision systems to recognize arbitrary FIFA balls recently [29–31]. Because their panoramic mirrors are hyperbolic, the balls are imaged to be circles in the panoramic images. So Martins et al. used canny operator to detect the edges, and then applied the circular Hough transform to detect all the candidate circles imaged by the balls [29,30]. An effective validation process was proposed to discard the false positives. Zweigle et al. used standard Hough transform to detect all the circles in the panoramic image, and then extracted the color histogram for each circle and compared it with the color histogram learned in the off-line calibration process to validate the real FIFA balls [31]. The off-line calibration step was needed in this method. Experimental results showed that the correct detection rates of these two methods were very high. However, all the experiments were performed in very clean environments.

3. Our omnidirectional vision system

3.1. The new panoramic mirror

The omnidirectional vision system consists of a convex mirror and a camera pointed upward towards the mirror. The characteristic of omnidirectional vision system is determined mostly by the shape of panoramic mirror. We design a new panoramic mirror which consists of the hyperbolic mirror, the horizontally isometric mirror and the vertically isometric mirror from the interior to the exterior. This mirror is different from our former one which only includes the horizontally isometric mirror and the vertically isometric mirror [32]. The deficiency of our former one is that the imaging of the scene which is very close to the robot is bad, such as the robot itself cannot be seen in the panoramic image, which is caused by the difficulty in manufacturing the innermost part

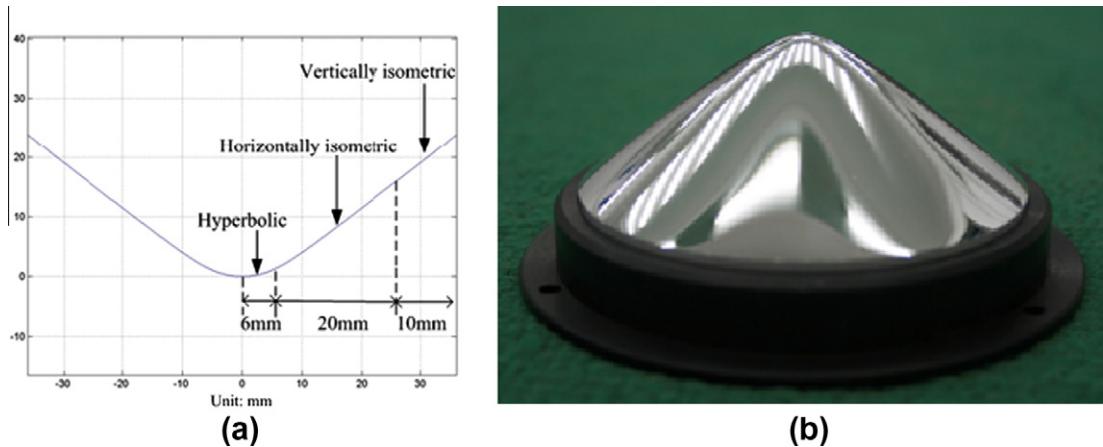


Fig. 1. Our new panoramic mirror. (a) The profile curve. (b) The manufactured panoramic mirror.

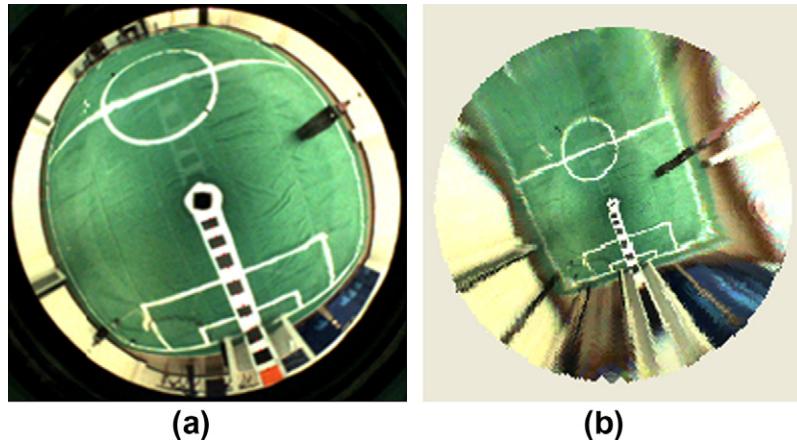


Fig. 2. (a) The panoramic image captured by our omnidirectional vision system. (b) The calibrated result by mapping the image coordinate to the real world coordinate.

of the mirror accurately. Replacing the innermost part with a hyperbolic mirror solves this problem. The designed profile curve of the new mirror and the manufactured mirror are demonstrated in Fig. 1a and b. The typical panoramic image captured by our new omnidirectional vision system is shown in Fig. 2a in a RoboCup MSL standard field with the dimension of 18 m * 12 m. The new omnidirectional vision system not only maintains the merit of our former system, which makes the imaging resolution of the objects near the robot on the field constant and the imaging distortion of the objects far from the robot small in the vertical direction, but also enables the robot to acquire a very clear imaging of the scene which is very close to it, such as the robot itself.

3.2. The calibration of our omnidirectional vision

Our omnidirectional vision system is not a single viewpoint one [33], therefore we use the model-free calibration method [34] proposed by Voigtländer et al. to calibrate it. First, we use the Canny operator to detect the edge information of the panoramic image, and obtain 15 edge points of the calibration patch shown in Fig. 2a as the support vectors in each pre-defined direction. Then we calculate the distance map for each point in the image by using Lagrange interpolation algorithm in the radial direction and rotary direction respectively. Fig. 2b shows the calibrated result by mapping the image coordinate to the real world coordinate.

4. Camera parameters auto-adjusting algorithm based on image entropy

In this section, we propose a novel method to auto-adjust the camera parameters based on image entropy to achieve the robustness and adaptability of the camera's output with respect to different lighting conditions. We define the camera parameters adjustment as an optimization problem, and use image entropy as the optimizing goal.

4.1. Image entropy

The setting of the camera parameters affects the quality of the acquired images greatly. Taking the camera of our omnidirectional vision system as an example, only exposure time and gain are able to be adjusted (auto white-balance has been achieved in the camera, so we do not consider white-balance). Three typical images acquired with different parameters are shown in Fig. 3. The quality of the images in Fig. 3a and c is much worse than that in Fig. 3b, because they are less-exposed and over-exposed respectively, and the image in Fig. 3b is well exposed. The two images in Fig. 3a and c cannot represent the environment well, and we can say that the information content of these two images are less than that in Fig. 3b. So both less-exposure and over-exposure will cause the loss of image information [35].

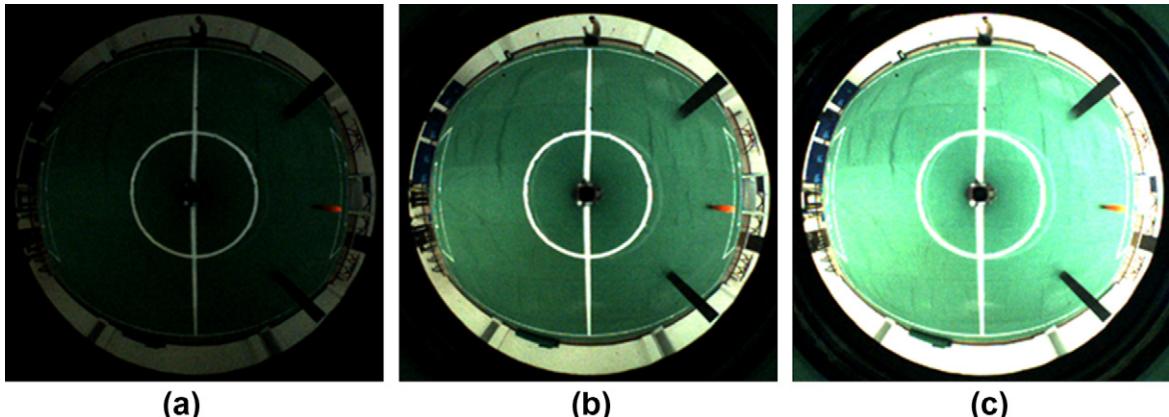


Fig. 3. The images acquired by our omnidirectional vision system with different exposure time. The gain is 18. The exposure time is: (a) 5 ms; (b) 18 ms; and (c) 40 ms.

According to Shannon's information theory [36], information content can be measured by entropy, and entropy increases with the information content. So we use image entropy to measure the image quality, and we will verify the relationship between the entropy of the acquired images and camera parameters by analyzing the distribution of image entropies with different camera parameters.

We use Shannon's entropy to define image entropy. Because RGB color space is a linear color space that formally uses single wavelength primaries and the color values are obtained directly after the CCD sensing of color cameras, it is more appropriate to calculate image entropy in RGB color space than in YUV or HSV color space. So the entropy of a color image is defined as follows:

$$E_c = - \sum_{Ri=0}^{L-1} P_{Ri} \log P_{Ri} - \sum_{Gi=0}^{L-1} P_{Gi} \log P_{Gi} - \sum_{Bi=0}^{L-1} P_{Bi} \log P_{Bi} \quad (1)$$

where $L = 256$ is the number of discrete levels of RGB color channels, and P_{Ri} , P_{Gi} , P_{Bi} are the probability of color value Ri , Gi , Bi existing in the three color channels of the image. The probability P_{Ri} , P_{Gi} and P_{Bi} can be replaced with the frequency approximately, so they are calculated according to the three histogram distributions of the image in RGB color channels. Assuming the number of pixels having color value Ri is N_{Ri} and the image contains N pixels, $P_{Ri} = N_{Ri}/N$. Similarly, P_{Gi} and P_{Bi} can also be calculated. More details about the image histogram can be found in [37]. Several similar definitions of image entropy in gray images were proposed in [6,21, 35,38], where image entropy was used to measure the image quality or the quality of image processing.

According to the definition in Eq. (1),

$$0 = \text{Min}(E_c) \leq E_c \leq \text{Max}(E_c) = -3 * \sum_{i=0}^{255} (1/256) \log(1/256) \\ = 16.6355,$$

and image entropy will increase monotonously with the degree of average distribution of color values.

4.2. The relationship between image entropy and camera parameters

We captured a series of panoramic images by using our omnidirectional vision system with different exposure time and gain in indoor environment and outdoor environment, and calculated image entropy according to Eq. (1) to see how image entropy varies with the camera parameters. The indoor environment is a standard RoboCup MSL field with the dimension of 18 m * 12 m, where the illumination is not only determined by artificial lights, but also can be influenced greatly by natural light through lots of windows. The

outdoor environment includes two blue patches and several components of the indoor environment such as a piece of green carpet, two orange balls and several black obstacles. All the experiments of this section were performed in these two environments. Furthermore, because the illumination in these two environments were totally different, and the dynamic response range of our cameras were limited, we used two omnidirectional vision systems in two robots with different iris settings in the two environments. The iris can only be adjusted manually.

In the experiment of indoor environment, the range of exposure time was from 5 ms to 40 ms, and the range of gain was from 5 to 22. The experiment was carried out in the evening, so the illumination was not affected by natural light. In the experiment of outdoor environment, the range of exposure time was from 1 ms to 22 ms, and the range of gain was from 1 to 22. The weather was cloudy, and the experiment was carried out in the midday. The minimal adjusting step of the two parameters was 1 ms and 1. We acquired one image with each group of parameters. The changes of image entropy with the different camera parameters are shown in Figs. 4 and 5 in the two experiments.

From Figs. 4 and 5, we can conclude that the manner in which image entropy varies with camera parameters is the same in the two experiments, and there is a ridge curve (the blue¹ curve in Figs. 4 and 5). Along the ridge curve, the image entropies are almost the same in each experiment, and there is not obvious maximal value. So which image entropy along the ridge curve indicates the best image, or whether all the images related to the image entropies along the ridge curve are good?

For the images are used to be processed to realize object recognition, self-localization or other robot vision tasks, we tested the quality of images by using the same color calibration result [39] learned from one image corresponding to a certain entropy on the ridge curve to segment the images corresponding to all the entropies along the ridge curve and detect the white line points using the algorithm proposed in [14]. These line points are very important for the robot's visual self-localization. The typical images along the ridge curve and the processing results in the two experiments are demonstrated in Figs. 6 and 7.

As shown in the two figures, the images are well segmented by the same color calibration result in each experiment, and the object recognition can be realized successfully for soccer robots. The same processing results are achieved in all the other images related to the image entropies along the ridge curve. So all these images are good for robot vision, and there is some kind of color

¹ For interpretation of color in all figures, the reader is referred to the web version of this article.

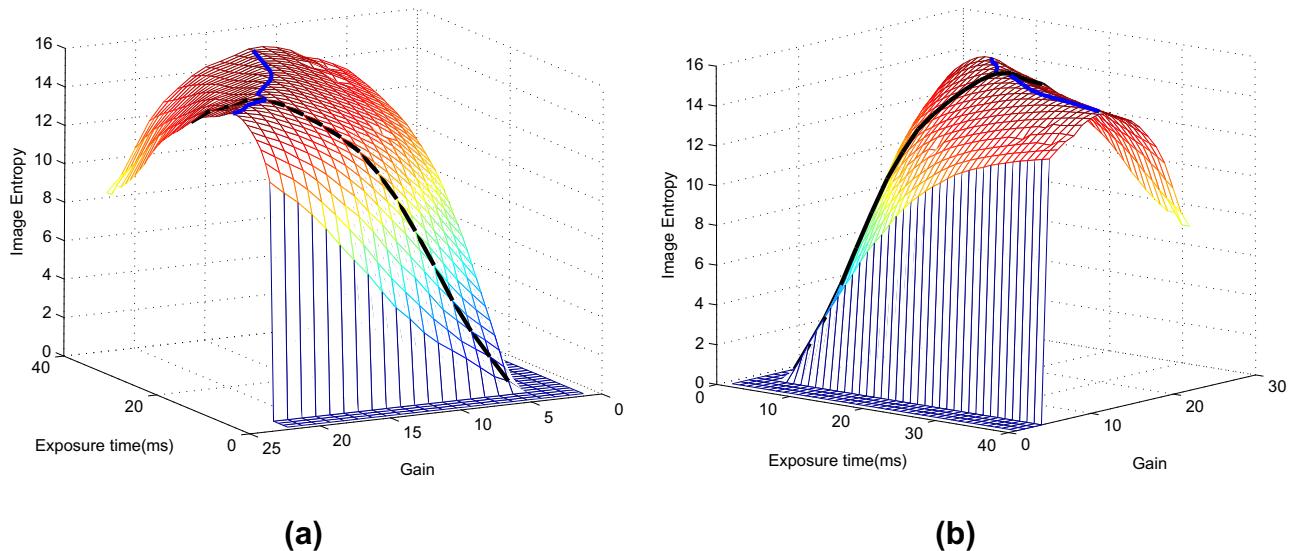


Fig. 4. The change of image entropy with different exposure time and gain in indoor environment. (a) and (b) are the same result viewed from two different view angles.

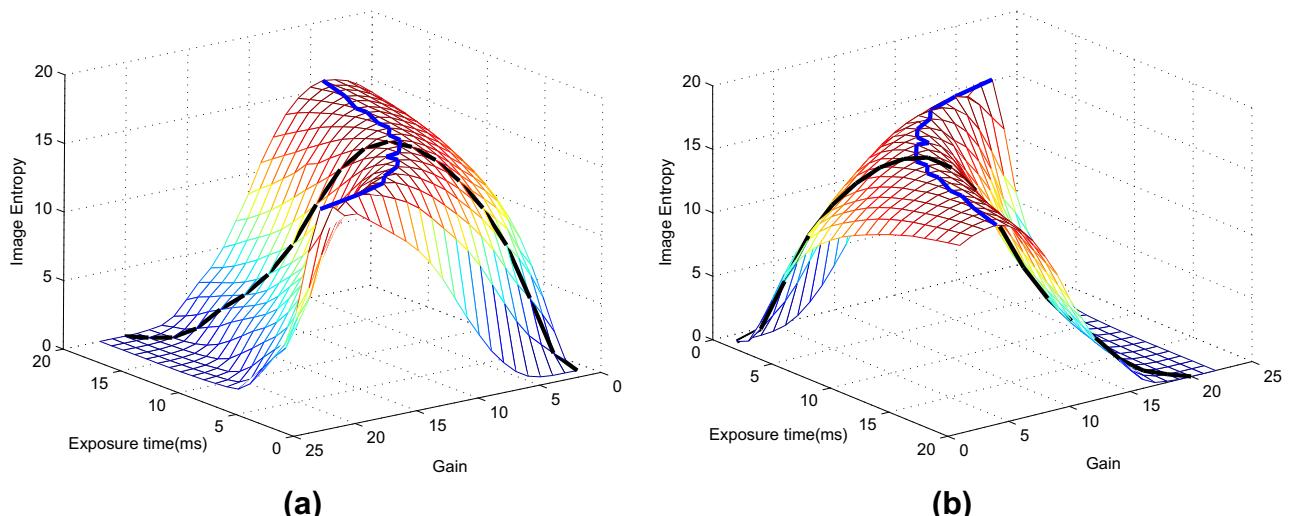


Fig. 5. The change of image entropy with different exposure time and gain in outdoor environment. (a) and (b) are the same result viewed from two different view angles.

constancy in these images, though they are acquired with different camera parameters. It also means that all the settings of exposure time and gain corresponding to the image entropies along the ridge curve are acceptable for robot vision. So it is verified that image entropy can indicate whether camera parameters are set properly.

4.3. Camera parameters auto-adjustment based on image entropy

According to the experiments and analysis in Section 4.2, image entropy can indicate the image quality for robot vision and whether camera parameters are set properly. Therefore, camera parameters adjustment can be defined as an optimization problem, and image entropy can be used as the optimizing goal. But as shown in Figs. 4 and 5, the image entropies along the blue ridge curve are almost the same, and so it is not easy to search for the optimal solution. Furthermore, the camera parameters themselves affect the performance of vision systems. The increase of exposure time will cause the decrease of real-time performance. The higher gain will bring more image noises which will cause worse results in color segmentation or other image processing. So exposure time

and gain have to be taken into account in this optimization problem. But it is difficult to measure the degree of these parameters' effect, making it almost impossible to add some indicative or constraint function to image entropy directly for the optimization problem.

Considering that all the images related to the image entropies along the ridge curve are good for robot vision, we turn the two-dimension optimization problem into a one-dimension one by defining some searching path. As mentioned above, the increase of exposure time and gain will worsen the performance of robot vision. So a compromise between exposure time and gain should be made, and neither of them should be too high. Considering the range of exposure time and gain of our cameras, we define the searching path as “exposure time = gain” (just equal in number value, for the unit of exposure time is millisecond, and there is not a unit for gain). We search for the maximal image entropy along this path, and the camera parameters corresponding to the maximal image entropy are best for robot vision in the current environment and under the current lighting condition. The searching path is shown as the black curve in Figs. 4 and 5 respectively in indoor

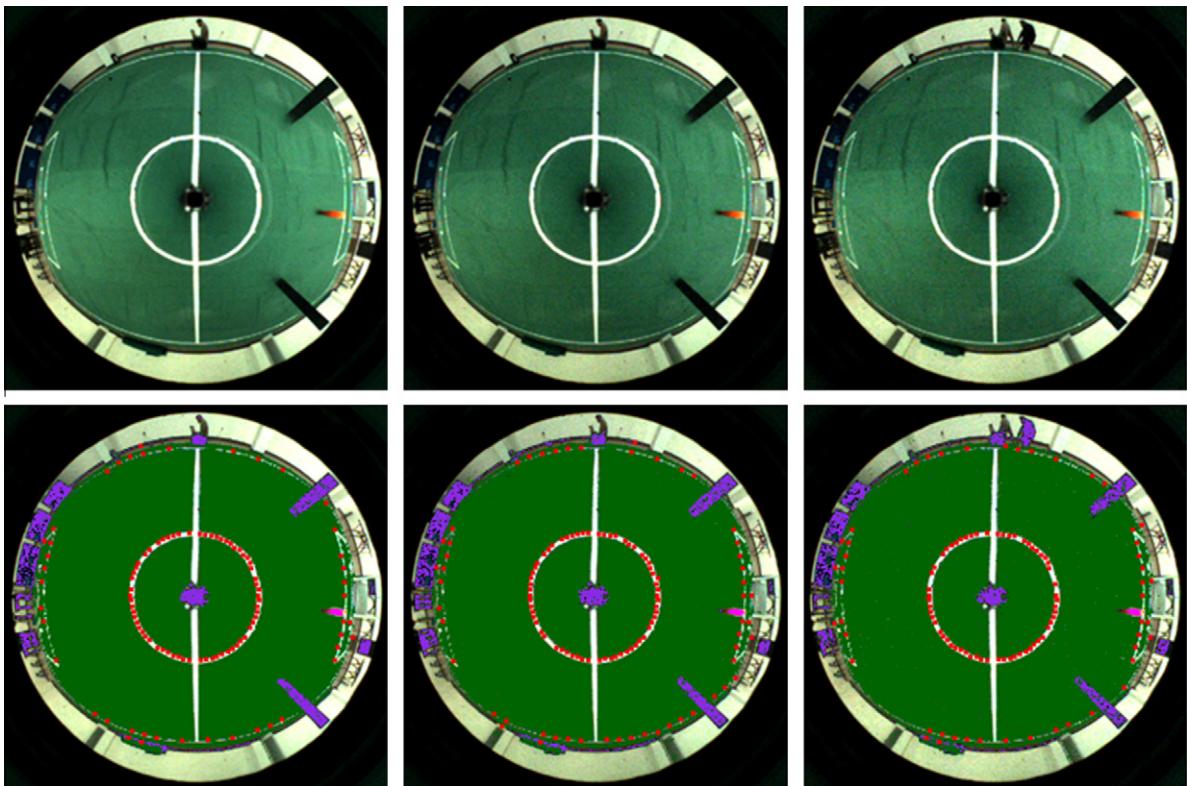


Fig. 6. (Top) The typical images along the ridge curve and (bottom) the processing results in indoor experiment. The red points are the detected white line points. The camera parameters are as follows: (left) exposure time: 34 ms, gain: 13; (middle) exposure time: 18 ms, gain: 18; (right) exposure time: 14 ms, gain: 21. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

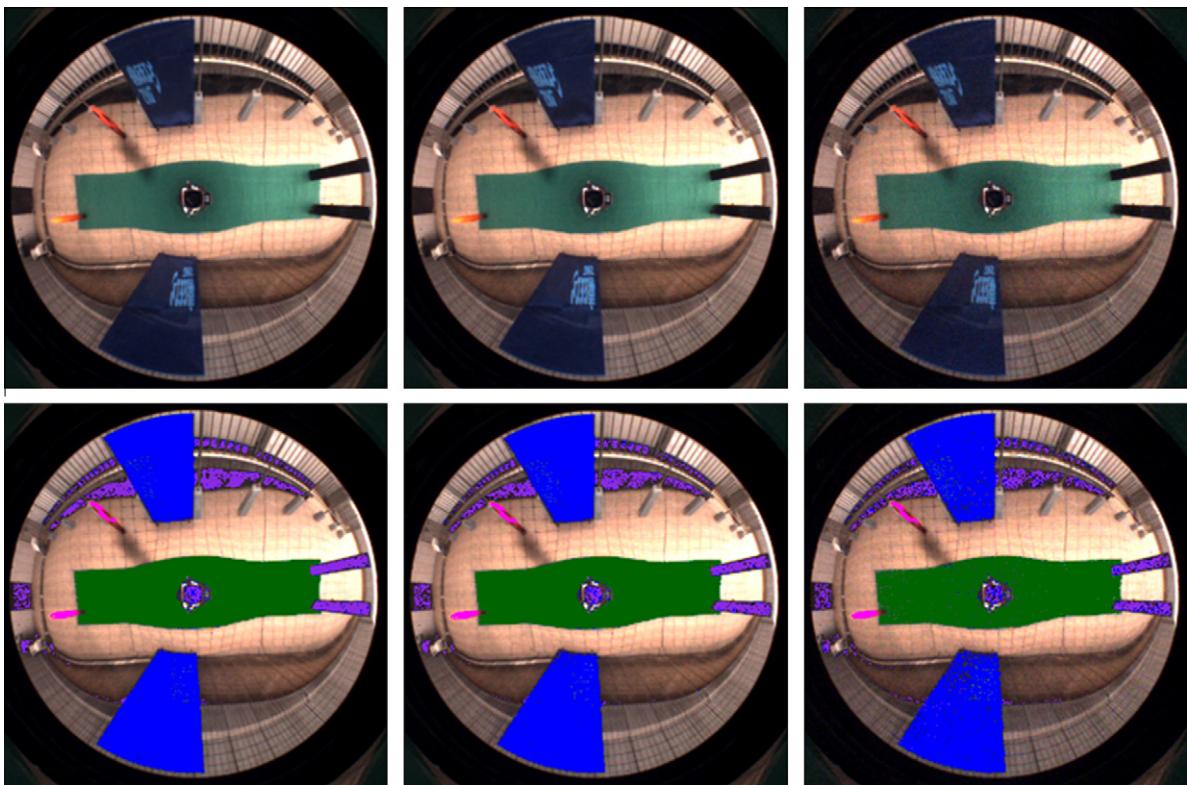


Fig. 7. (Top) The typical images along the ridge curve and (bottom) the processing results in outdoor experiment. In this experiment, there are not white line points to detect. The camera parameters are as follows: (left) exposure time: 17 ms, gain: 5; (middle) exposure time: 9 ms, gain: 9; (right) exposure time: 2 ms, gain: 18.

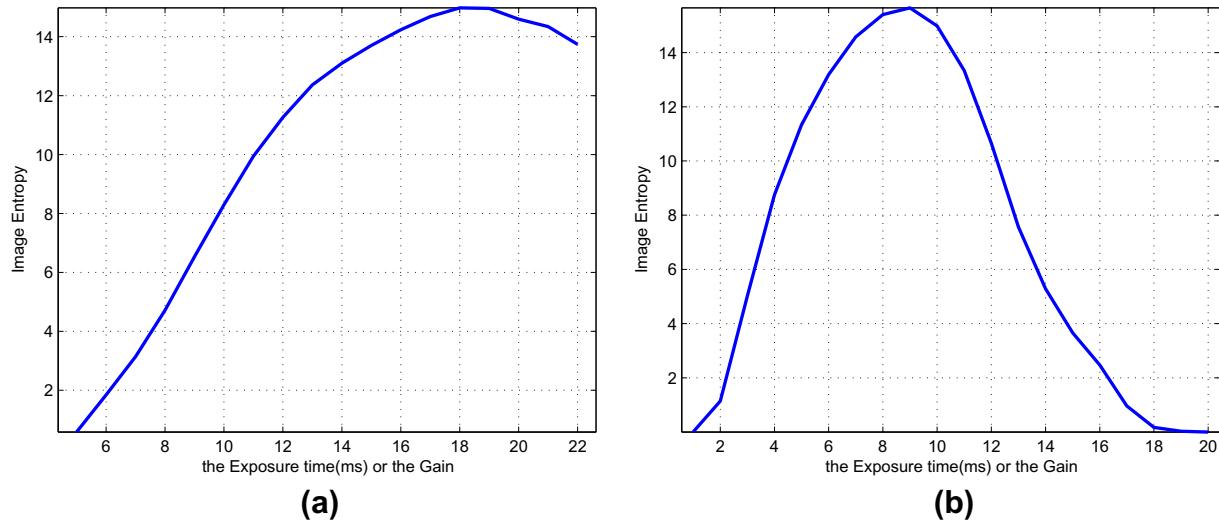


Fig. 8. The distribution of image entropy along the defined searching path “exposure time = gain” in indoor environment (a) and outdoor environment (b).

environment and outdoor environment. The distributions of image entropy along the path in the two environments are demonstrated in Fig. 8.

From Fig. 8, a very good characteristic of image entropy is that image entropy will increase monotonously to the peak and then decrease monotonously along the defined searching path. So the global maximal image entropy can be found easily by searching along the defined path, and the best camera parameters are determined at the same time. In Fig. 8a, the best exposure time and gain are 18 ms and 18 respectively; in Fig. 8b, the best exposure time and gain are 9 ms and 9.

According to the special characteristic of the omnidirectional vision, the robot itself is imaged in the central region of the panoramic image. Therefore in the actual application, the robot can judge that whether it has entered into a totally new environment or if the illumination has changed in the current environment by calculating the mean brightness value in the central part of the panoramic image. If the increase of the mean value is higher than a certain threshold, the illumination becomes stronger, and the optimization of camera parameters should be run in the direction that exposure time and gain reduce along the searching path. Similarly, if the decrease of the mean value is higher than the threshold, the optimization should be run in the direction that exposure time and gain raise along the searching path. In our experiment, we set the threshold as 20. The mean value will not be considered during the optimizing process, and it is not needed either. After the optimal camera parameters have been achieved, it will be calculated and saved again, and it will be compared with the new ones to judge whether the adjustment of the camera parameters is needed. Of course, this judgement can be scheduled to perform only once every several cycles, because the changes in illumination are not that rapid.

When the robot is in different positions on the field, the acquired images will be different, so the maximal image entropy along the searching path varies with the different locations of the robot. Therefore, once the robot finds that camera parameters auto-adjustment is needed, it should stop at the same place till the optimization is finished. Furthermore, according to the discussion in Section 4.5, camera parameters optimization can be finished at most in several hundred milliseconds, so the robot's surrounding can be considered to be static approximately during the optimizing process. Image entropy only changes with different camera parameters, and the optimization will not be affected by the robot's surrounding.

In the optimizing process, a new group of parameters are set into the camera, and then a new image is acquired and image entropy can be calculated according to Eq. (1). The new entropy is then compared with the last one to check whether the maximal entropy has been reached. This iteration continues until the maximal entropy has been reached. Regarding how to choose the new parameters, the technique of varying optimizing step is used to accelerate the optimizing process. When the current image entropy is not too far from $\text{Max}(E_c)$, the optimizing step can be 1, which means that the change of exposure time is 1 ms and the change of gain is 1. When the current image entropy is too far from $\text{Max}(E_c)$, the optimizing step can be 2 or 3.

The searching path should be changed according to different camera characteristics, and the different requirements of the vision system in different applications. For example, the value range of gain depends on the camera used. Some range is from 0 to 50, and others may be from 0 to 4000. So the searching path should be “exposure time = $\alpha * \text{gain}$ ” (also just equal in number value). The variable α can be determined manually after analyzing the gain’s effect on the images. For the cameras with similar parameter ranges as ours, in some applications, the signal to noise ratio of the image is required to be high and the real-time performance is not necessary, so the searching path can be “exposure time = $\alpha * \text{gain}$ ”, with $\alpha > 1$. In some other cases, the camera is required to output images as quickly as possible and the image noise is not restricted too much, so the searching path can be “exposure time = $\alpha * \text{gain}$ ”, with $\alpha < 1$.

4.4. The experimental results

We tested our camera parameters auto-adjusting algorithm under different lighting conditions in indoor environment and outdoor environment respectively. We verified whether camera parameters were set properly by processing the images using the same color calibration result learned in the experiments of Section 4.2. We also evaluated the robot's self-localization based on the omnidirectional vision after camera parameters had been optimized under different lighting conditions.

4.4.1. The experiment in indoor environment

In this experiment, the weather was cloudy, and the time was midday, so the illumination was influenced by both artificial and natural light. We changed the illumination by gradually turning off some lamps. We used the color calibration result in the indoor

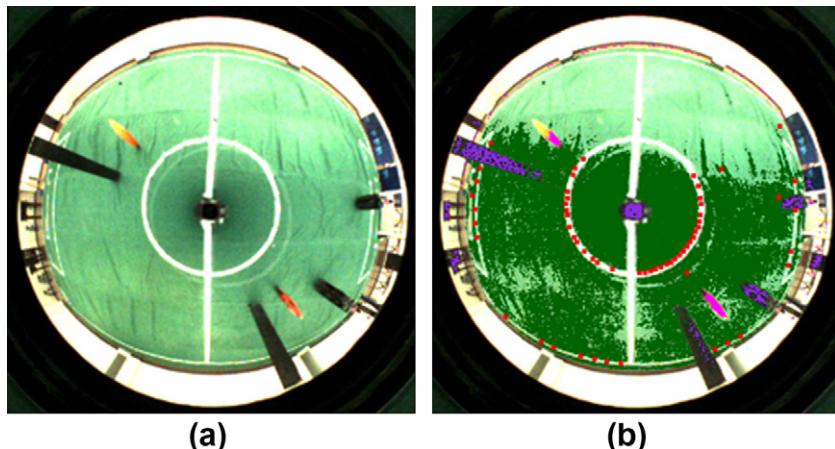


Fig. 9. The acquired image (a) and the image processing result (b) when camera parameters had not been optimized in indoor environment. The best parameters in Section 4.3 were used.

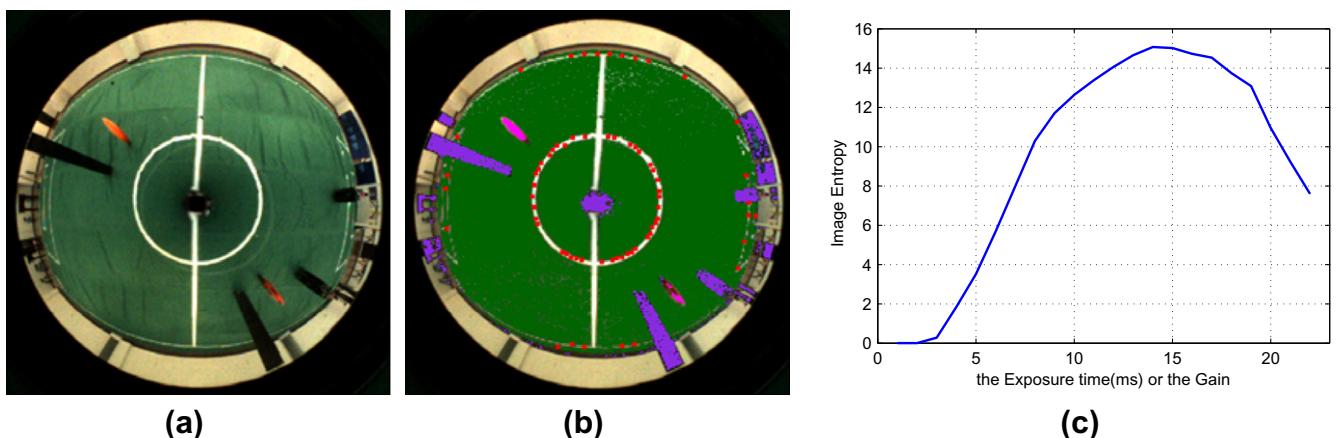


Fig. 10. (a) The acquired image after camera parameters had been optimized in indoor environment. (b) The image processing result. (c) The distribution of image entropy along the searching path "exposure time = gain".

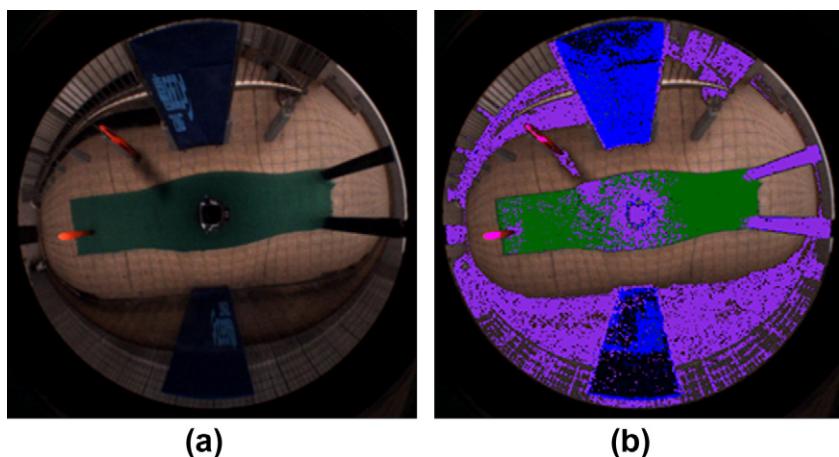


Fig. 11. The acquired image (a) and the image processing result (b) when camera parameters had not been optimized in outdoor environment. The best parameters in Section 4.3 were used.

experiment of Section 4.2 to process the images for soccer robots. When all the lamps were on and the camera was set with the best parameters in Section 4.3, the acquired image and the processing result are shown in Fig. 9. The image was over-exposed, and it

could not be processed well. After the parameters had been optimized by our method, the acquired image and the processing result are demonstrated in Fig. 10a and b. The distribution of image entropy along the searching path is shown in Fig. 10c. The

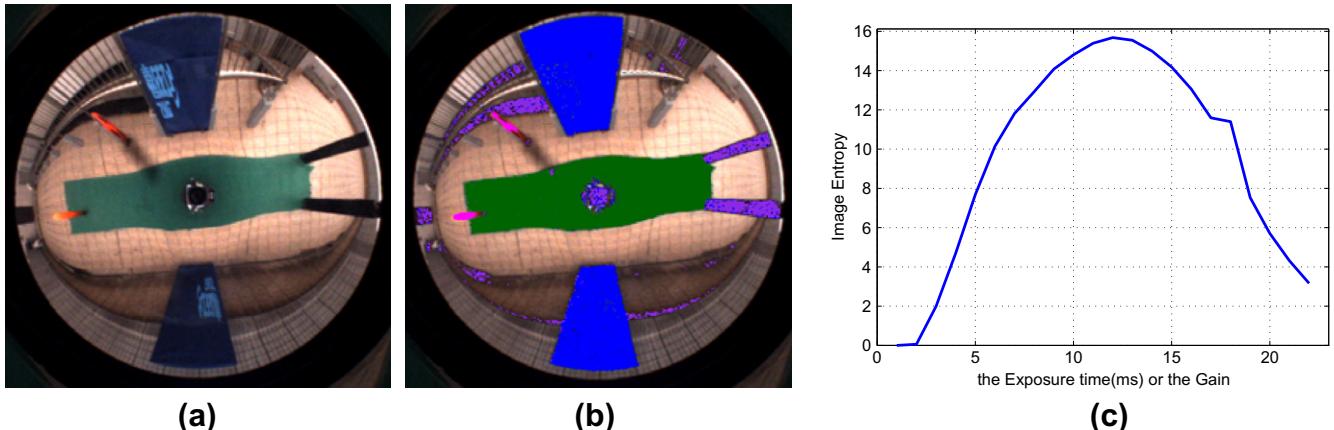


Fig. 12. (a) The acquired image after camera parameters had been optimized. (b) The image processing result. (c) The distribution of image entropy along the searching path “exposure time = gain”.

optimal exposure time was 14 ms and gain was 14. The image was well-exposed, and the processing result was also good. When the illumination changed gradually, similar results were achieved.

4.4.2. The experiment in outdoor environment

In this experiment, the weather was sunny, and the time was from midday to dusk, so the illumination was from brightness to darkness determined by natural light. We also used the same color calibration result in the outdoor experiment of Section 4.2 to process the images for soccer robots. The acquired image and the pro-

cessing result are shown in Fig. 11, when the time of the experiment was at 16:00, and the camera was set with the best parameters in Section 4.3. The image was less-exposed, and the processing result was unacceptable for robot vision. After the parameters had been optimized, the acquired image and the processing result are demonstrated in Fig. 12a and b. The distribution of image entropy along the searching path is shown in Fig. 12c. The optimal exposure time was 12 ms and gain was 12. The image was well-exposed, and the processing result was good. We also processed the images acquired with some suboptimal camera param-

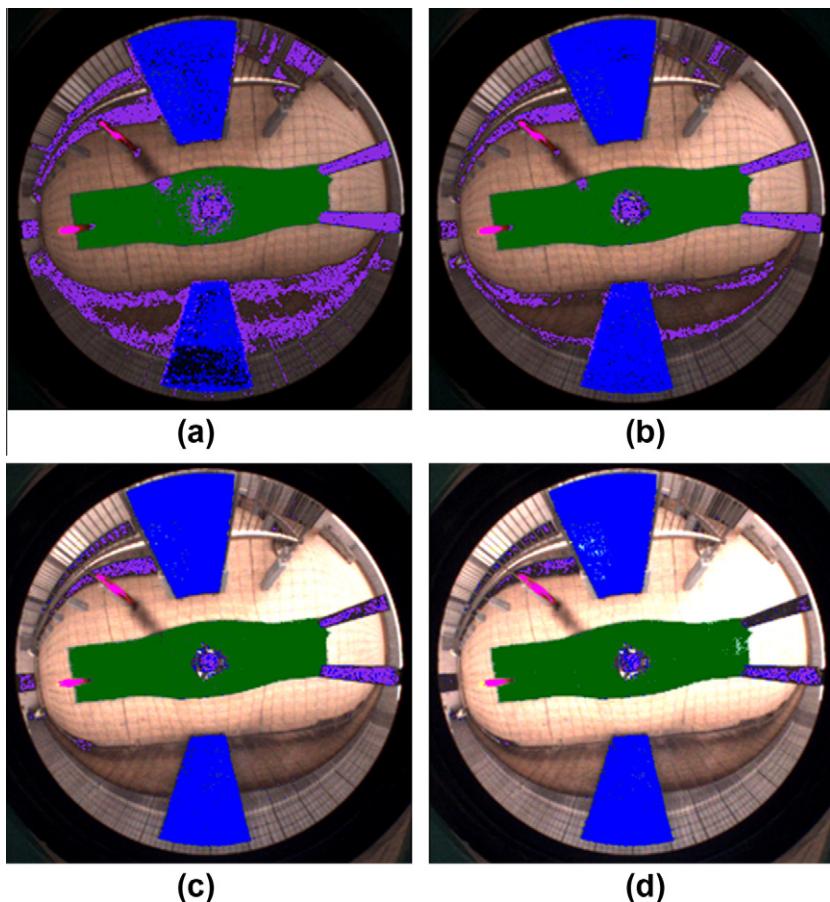


Fig. 13. The processing results of the images acquired with some suboptimal camera parameters in outdoor environment. (a) Exposure time: 10 ms, gain: 10; (b) exposure time: 11 ms, gain: 11; (c) exposure time: 13 ms, gain: 13; (d) exposure time: 14 ms, gain: 14.

eters, and those results are demonstrated in Fig. 13. All the color classification results in Fig. 13 are more or less worse than that in Fig. 12. This verifies that the image captured with the optimal camera parameters is the optimal image for robot vision. When the experiments were run in different times from midday to dusk, all of the images were well-exposed and well processed after camera parameters had been optimized.

4.4.3. Evaluation of robot's localization under different lighting conditions

In this experiment, we evaluated the robot's self-localization results based on the omnidirectional vision with optimized camera parameters under very different lighting conditions in three cases. In the first case, the lighting condition was the same as that in the experiment of Section 4.3. In the second case, the illumination was affected greatly by sun rays in a sunny day, and the optimal exposure time and gain were 12 ms and 12 respectively. The acquired image and the processing result are shown in Fig. 14a and b. The distribution of image entropy along the searching path is shown in Fig. 14c. In the third case, the weather and the time of the experiment were similar to those in Section 4.4.1, but we changed the illumination dynamically during robot's localization process by turning the lamps on and off, so the camera parameters would be auto-adjusted in real-time when the robot detected that the illumination changed.

The robot's self-localization results by the method proposed in [40] in the three cases are demonstrated in Fig. 15. During the experiment, the robot was pushed by people to follow some straight tracks on the field shown as the black lines in Fig. 15. The red traces depict the robot's self-localization results. The statistics of localization errors is shown in Table 1. The robot was able to achieve good localization results with the same color calibration result under very different and even dynamic lighting conditions, though sometimes the effect from sun rays was so strong that the maximal localization error in the second case was much larger. If camera parameters were not adjusted according to the changes of the illumination, the robot's self-localization would fail when using the same color calibration result in the latter two cases. This experiment also verifies that our camera parameters auto-adjusting method is effective.

4.5. Discussion

According to the analysis and the experimental results in the above subsections, our camera parameters auto-adjusting method

based on image entropy can make the camera's output adaptive to different lighting conditions and describe the real world as consistently as possible. So some kind of color constancy for our omnidirectional vision is achieved. Robust object recognition and self-localization can be achieved in a color-coded MSL environment under varying lighting conditions for soccer robots. Furthermore, unlike other existing methods, there are not any referenced colors needed during the optimizing process, so our method can be applied in many more situations. Our method also provides an objective setup technique for vision/camera parameters, and the users don't need to adjust them manually according to experiences when the robots come into a new working environment.

Besides exposure time and gain adjusted in the above experiments, our method can be extended to adjust more parameters if supported by hardware. We replaced the original lens of our perspective camera with the HZC08080 lens (the size of HZC08080 lens is too large to be mounted in the omnidirectional vision system), so the iris can be adjusted by sending commands to control the motors of the lens in software. The distribution of image entropy with different iris and exposure time, the image entropies along the defined searching path and the optimal image along this path are shown in Fig. 16.

Besides omnidirectional vision systems, our method can also be applied to any other vision system, but it is not easy for other vision systems to judge whether camera parameters adjustment is needed, for there is no obvious region existing constantly in their field of view like omnidirectional vision systems. Maybe some special object could be recognized and tracked, and its image region can be used as reference to judge whether the illumination changes.

About the real-time performance of our method, for the lighting condition will not change so suddenly in actual applications, it takes only several cycles to finish the optimizing process. And it takes about 40 ms to set the parameters into our camera each time. So camera parameters adjustment can be finished at most in several hundred milliseconds, and there is no problem in the real-time requirement to adjust camera parameters occasionally during the competition.

However, there are still some deficiencies in our algorithm. For example, it cannot deal with the situation that the illumination is highly nonuniform. Entropy is a global appearance feature for images, so it may be not the best optimizing goal in this situation. As shown in Fig. 17, although camera parameters have been optimized as 21 ms and 21, the image processing result is still unacceptable for robot vision. Maybe object recognition or tracking

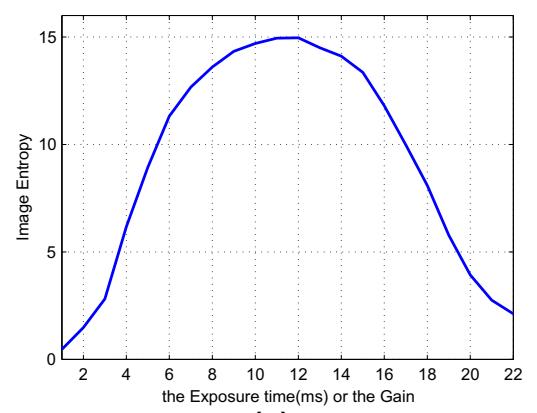
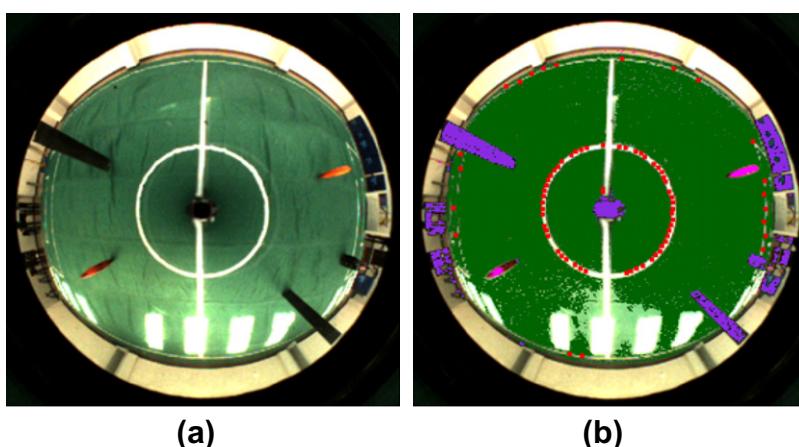


Fig. 14. (a) The acquired image after camera parameters had been optimized. The illumination was affected greatly by sun rays through windows. (b) The image processing result. (c) The distribution of image entropy along the searching path “exposure time = gain”.

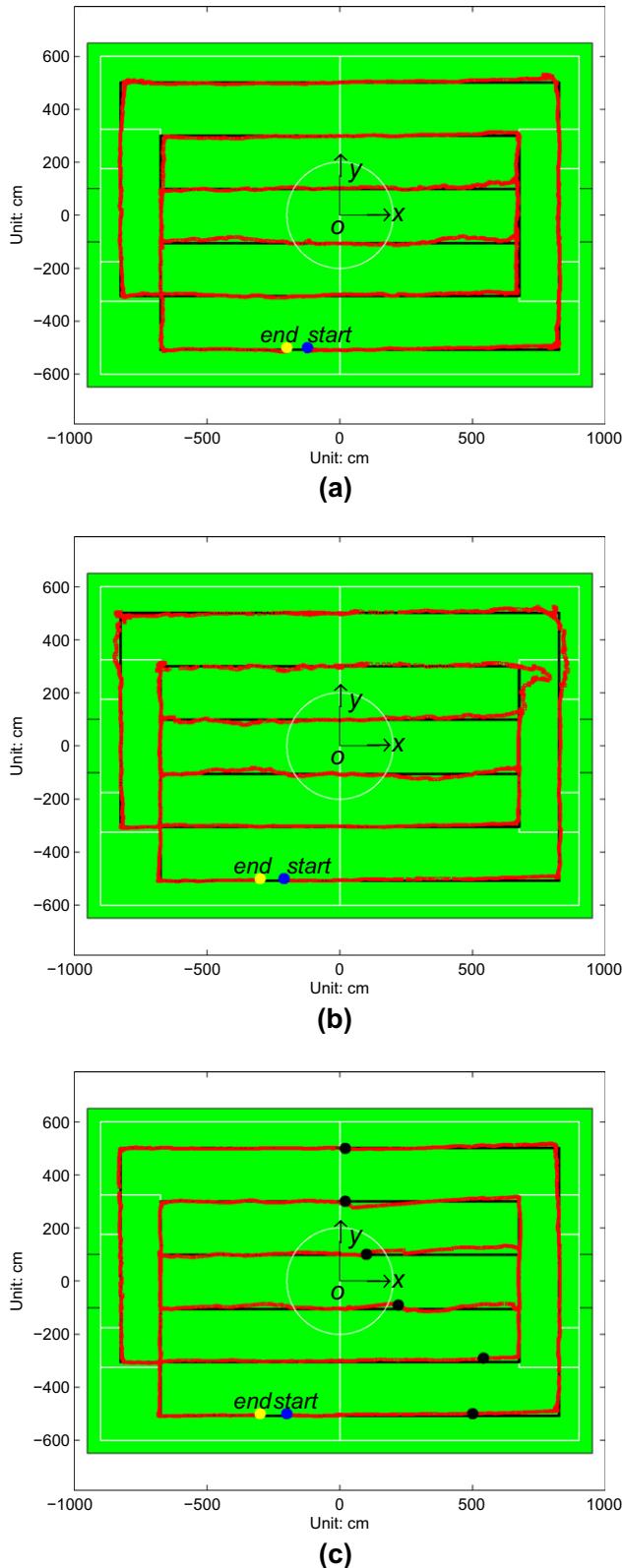


Fig. 15. The robot's localization results after camera parameters had been optimized under totally different lighting conditions. (a) The illumination was not affected by natural light. (b) The illumination was affected greatly by sun rays. (c) The illumination changed dynamically. The black points are the positions where the illumination changed and camera parameters were auto-adjusted.

techniques should be integrated into our method, so camera parameters can be optimized according to the local features in the object region.

Table 1

The statistics of robot's self-localization errors under different lighting conditions. In this table, x , y , and θ are the self-localization coordinates related to the location x , y and orientation.

	x (cm)	y (cm)	θ (rad)
<i>In the first case</i>			
Mean error	5.907	5.967	0.044
Standard dev	7.334	7.117	0.052
Maximal error	30.724	35.595	0.286
<i>In the second case</i>			
Mean error	6.416	5.544	0.067
Standard dev	12.431	7.381	0.093
Maximal error	95.396	33.063	0.580
<i>In the third case</i>			
Mean error	2.751	5.867	0.047
Standard dev	3.593	7.533	0.061
Maximal error	16.834	35.173	0.279

5. Ball recognition based on the omnidirectional vision without color classification

In this section, we propose a ball recognition method based on the omnidirectional vision without color classification. We will analyze the imaging characteristic of the ball in our omnidirectional vision system, and then present the arbitrary FIFA ball recognition algorithm for soccer robots. The experiments will be performed in cluttered environments to test our algorithm.

5.1. Analysis of the imaging characteristic of the ball

In order to analyze the imaging characteristic of the ball in our omnidirectional vision system, we consider the situation that the ball is located on the ground, and assume that the panoramic mirror is a point with height h above the ground, for the mirror size is far smaller when compared to the ball size and the distance from the mirror to the ball. Therefore, the incident rays from the ball to the mirror can be said to form a cone tangent to the ball approximately. The intersections of a plane and a cone generate conic sections, such as circles, ellipses, hyperbolas, and parabolas. In this situation, an ellipse is generated by the intersection of the cone and the ground plane, which is shown in Fig. 18. We define a right hand Cartesian coordinate with the center of the robot on the plane as the origin o of the coordinate, with the direction from the robot to the ball on the plane as x axis. The direction of the major-axis of the ellipse coincides with x axis. We assume that the distance between the ball and the robot is x_b . The imaging of the ball is the same as the imaging of the ellipse in the omnidirectional vision system, so we need to derive the shape parameters of the ellipse on the ground and then analyze what shape the ellipse will be imaged in the panoramic image. The equation of the ellipse is as follows:

$$\frac{(x - x_0)^2}{a^2} + \frac{y^2}{b^2} = 1 \quad (2)$$

In Eq. (2), x_0 determines the location of the ellipse, a and b are the semi-major-axis and semi-minor-axis that determine the shape of the ellipse.

According to Fig. 18, we get the following equations:

$$x_b = x_c * (h - r)/h \quad (3)$$

$$d_b = \sqrt{(h - r)^2 + x_b^2} \quad (4)$$

$$d_s = \sqrt{d_b^2 - r^2} \quad (5)$$

$$\tan \theta_1 = r/d_s \quad (6)$$

$$\tan \theta = x_b/(h - r) \quad (7)$$

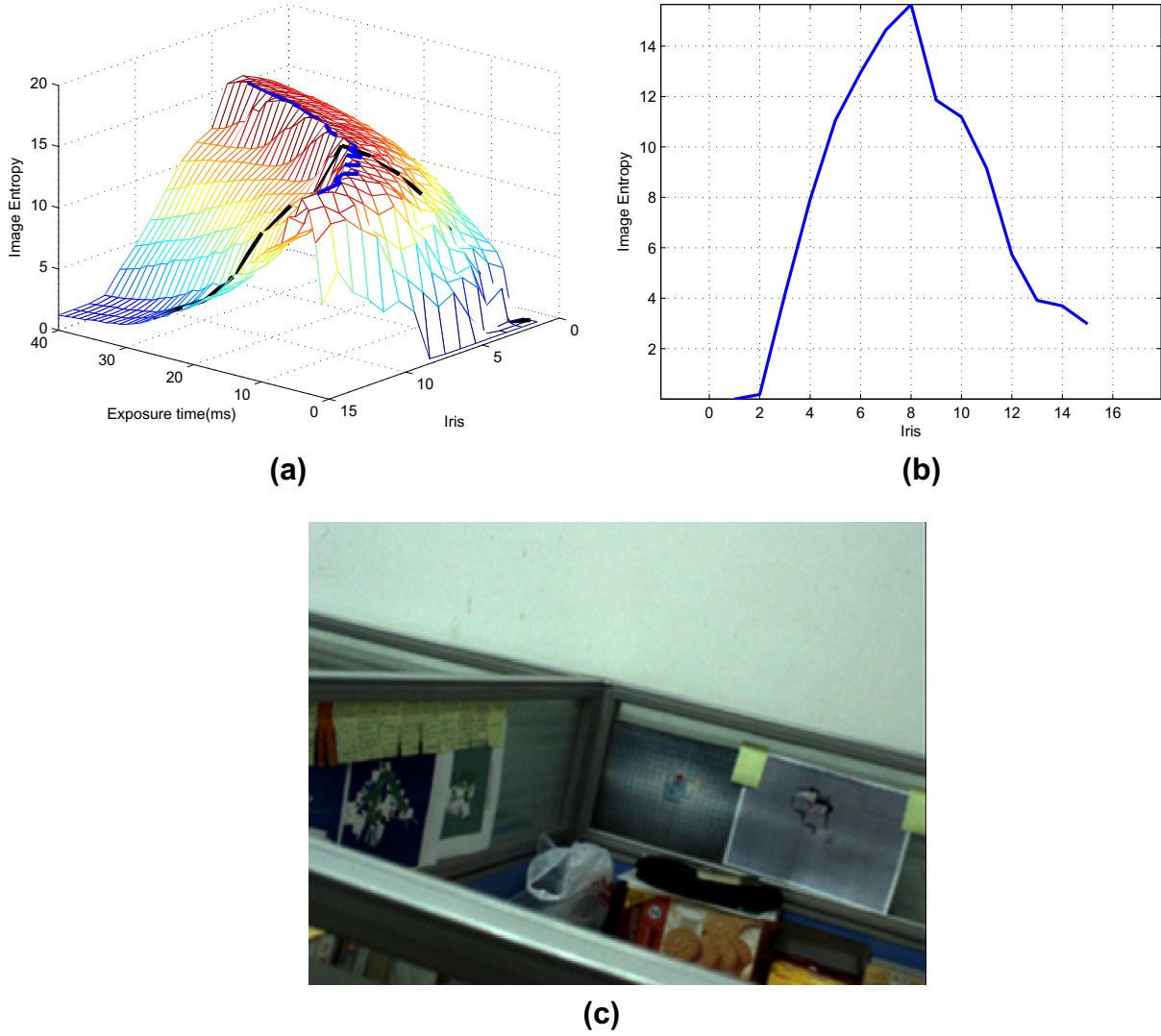


Fig. 16. (a) The distribution of image entropy with different iris and exposure time. (b) The image entropies along the defined searching path “exposure time = 1.73 * iris”. (c) The optimal image corresponding to the maximal entropy along the searching path.

$$\tan(\theta + \theta_1) = (\tan \theta + \tan \theta_1) / (1 - \tan \theta \tan \theta_1) \quad (8)$$

$$\tan(\theta - \theta_1) = (\tan \theta - \tan \theta_1) / (1 + \tan \theta \tan \theta_1) \quad (9)$$

$$x_l = h * \tan(\theta - \theta_1) \quad (10)$$

$$x_h = h * \tan(\theta + \theta_1) \quad (11)$$

$$d_k = \sqrt{h^2 + x_c^2} \quad (12)$$

$$y_c = d_k * \tan \theta_1 \quad (13)$$

$$a = (x_h - x_l) / 2 \quad (14)$$

$$x_0 = (x_h + x_l) / 2 \quad (15)$$

The height h of the mirror to the ground and the radius r of the ball are known in advance. If x_b or x_c is given, we can calculate a and x_0 by substituting Eqs. (4)–(11) into Eqs. (14) and (15). For the point (x_c, y_c) is located on the ellipse, we can derive the following equation:

$$\frac{(x_c - x_0)^2}{a^2} + \frac{y_c^2}{b^2} = 1 \quad (16)$$

We can derive b by substituting Eqs. (3), (12)–(15) into Eq. (16). In order to process the panoramic image to detect the ball, we have to derive further what shape this ellipse will be imaged in the im-

age. We assume that the ellipse will still be imaged as ellipse, and the distance between the center of the ellipse and the center of the ball is i in the panoramic image. According to the distance map of our omnidirectional vision system calibrated in Section 3, we can calculate $x_0 = f(i)$, where $f(\cdot)$ is the calibrated distance map function. It is very complex to calculate a and b if only x_0 is given according to Eqs. (3)–(16), so we mildly simplify this problem by replacing x_0 with x_c , for the point C is very close to the center of the ellipse in Fig. 18b. The simplification will be proved to be feasible by the experiments. So we can calculate $x_c = f(i)$, and then derive the real world ellipse parameters x_b , x_h , x_0 , a and b from x_c according to Eqs. (3)–(16). Because the distance map has already been calibrated in Section 3, we can use the inverse function of the distance map function to derive the semi-major-axis a_i and the semi-minor-axis b_i of the imaged ellipse in the panoramic image. The calculation functions are as follows:

$$a_i = (f^{-1}(x_h) - f^{-1}(x_l)) / 2 \quad (17)$$

$$b_i = b * f^{-1}(x_0) / x_0 \quad (18)$$

Up to now, we have finished the derivation on the shape parameters a_i and b_i of the ellipse imaged by the ball on the field ground, given that the distance between the center of the ellipse and the

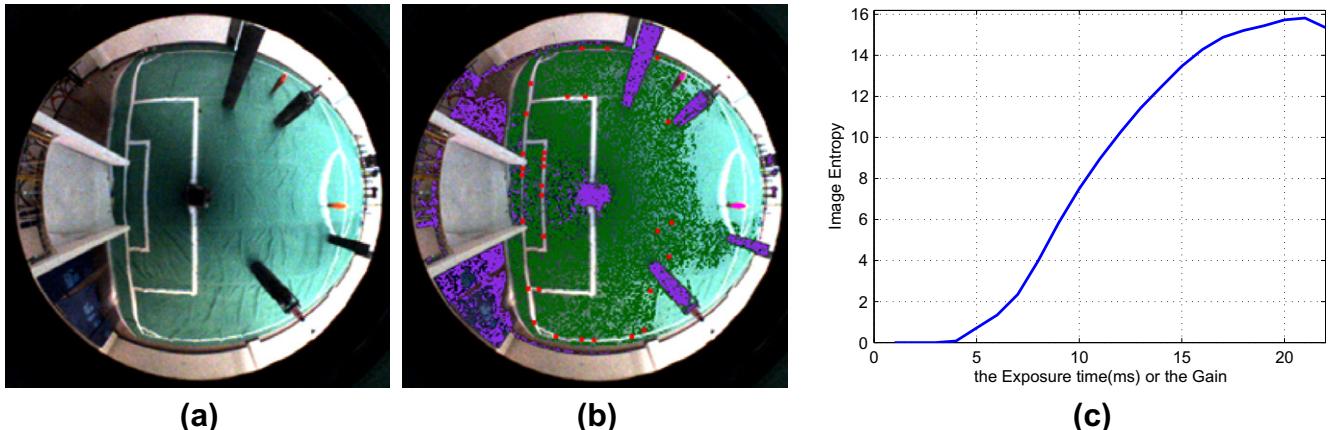


Fig. 17. The deficiency of our camera parameters auto-adjusting method. (a) The acquired image after camera parameters have been optimized when the illumination is highly nonuniform and the robot is located in a very dark place. (b) The image processing result. (c) The distribution of image entropy along the searching path “exposure time = gain”.

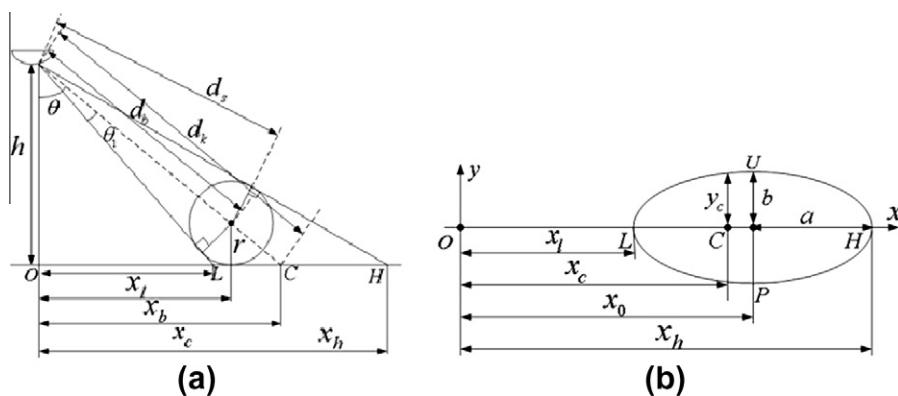


Fig. 18. The sketch of the imaging of the ball in the omnidirectional vision system. (a) The front view of the imaging of the ball. (b) The ellipse generated by the intersection of the cone and the ground plane.

center of the robot is i in the panoramic image. The process of the calculation can be summarized as follows: $i \rightarrow x_c \rightarrow x_l, x_0, x_h, a, b \rightarrow a_i, b_i$.

We store all the values of a_i and b_i varying with i in a lookup table which will be used to detect the arbitrary FIFA ball in the following image processing algorithm.

5.2. Image processing algorithm for ball recognition

We have derived the semi-major-axis and the semi-minor-axis of the ellipse imaged by the ball on each location of the panoramic image, so we can recognize the arbitrary ball by searching for the possible ellipses in the image according to this characteristic. From Figs. 20 and 21, we see that the ellipses are small in the images. According to our experiment, the edge information of the ellipses cannot be detected well, and the current Hough transform algo-

rithms cannot work well to detect these ellipses. Therefore we develop our own image processing algorithm to solve this problem.

The arbitrary FIFA balls have different colors, so we cannot detect the ball based on color classification like the traditional color object recognition methods. However, color variations still exist between the pixels belonging to the two sides of the ball contour. So we define two color variation scan methods to detect the possible contour points. The first scan method is called rotary scan, in which we define a series of concentric circles with the center of the robot in the image as their common centers and we will do the following scan in the concentric circles one by one. In each concentric circle, we search the color variation of every two neighboring pixels, and the color variation is measured by Euclidean distance in YUV color space. If the color variation is higher than a threshold (we set it as 18), a possible contour point is found. We calculate the distance d between every two possible contour points on the same concentric circle, the middle point P of the same two possible contour points, and the distance i between P and the center of the robot in the image. The semi-minor-axis b_i of the ellipse with its center located on P can be acquired from the lookup table calculated in Section 5.1. If $d \approx 2 * b_i$, we take the point P as a possible ellipse center point.

The other scan method is called radial scan, in which we define 360 radial scan rays with the center of the robot in the image as their common origins, and we will do the following scan along the radial scan rays one by one. In each radial scan ray, we search the same color variation of every two neighboring pixels as in the rotary scan. If

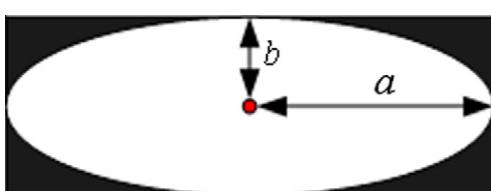


Fig. 19. The image regions inside and outside the ellipse for calculating the color difference.

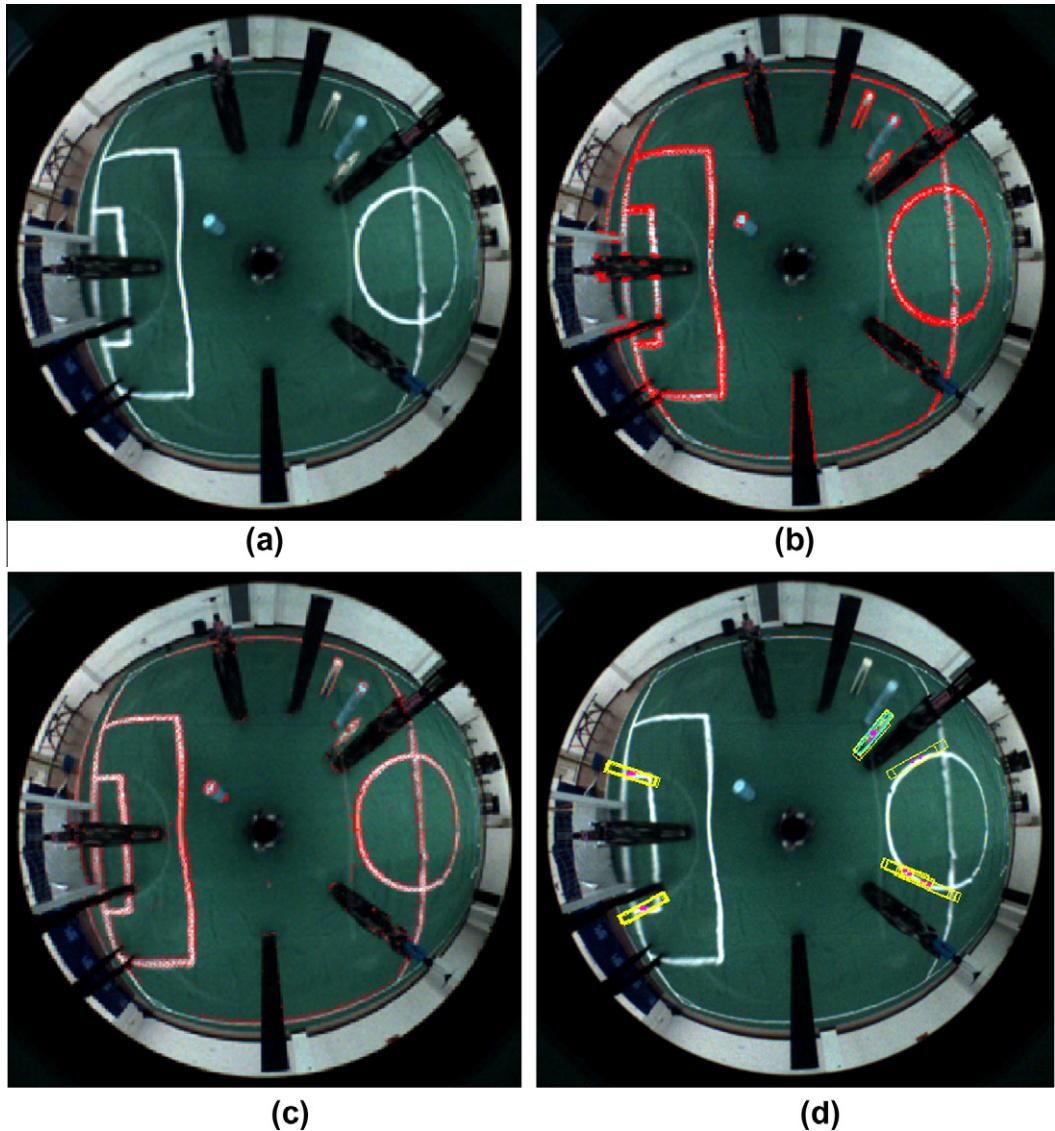


Fig. 20. The process and the results of our ball recognition algorithm. (a) The panoramic image. (b) The possible contour points detected by rotary scan. (c) The possible contour points detected by radial scan. (d) The ball recognition result.

the color variation is higher than a threshold (we set it as 16), a possible contour point is found. We calculate the distance d between every two possible contour points on the same radial scan ray, the middle point P of the same two possible contour points, and the distance i between P and the center of the robot in the image. The semi-major-axis a_i of the ellipse with its center located on P can be acquired from the lookup table calculated in Section 5.1. If $d \approx 2 * a_i$, we take the point P as a possible ellipse center point.

After the two sets of the possible ellipse center points have been obtained by the rotary scan and the radial scan, we compare all the points in one set with all the points in another set one by one. If the two points almost coincide with each other, we will consider that a candidate ellipse exists with the coinciding point as its center and also get the equation of this candidate ellipse. Then we can find the real ball within these candidate ellipses by searching for the maximal color difference inside and outside the candidate ellipses. We define color difference as D_c , and it is calculated as follows:

$$D_c = |\bar{y}_i - \bar{y}_o| + |\bar{u}_i - \bar{u}_o| + |\bar{v}_i - \bar{v}_o| \quad (19)$$

In Eq. (19), $\bar{y}_i, \bar{u}_i, \bar{v}_i$ are the mean values of those pixels inside the ellipse in Y, U, V color channel respectively, and $\bar{y}_o, \bar{u}_o, \bar{v}_o$ are the mean values of those pixels outside the ellipse and inside the

rectangle tangent to the ellipse in Y, U, V color channel respectively. The image regions used to calculate these mean values are shown as the white and the black regions in Fig. 19. If the maximal color difference is higher than the threshold we set as 50, we can validate the corresponding candidate ellipse as the real one imaged by the ball.

In this image processing algorithm, there are three thresholds needed to be set. We determined them through experiments, in which we used several balls with different colors and textures, as shown in Fig. 21. We kept adjusting the thresholds until the number of the balls recognized was maximized, so the best thresholds were determined. Furthermore, the camera parameters auto-adjusting algorithm proposed in Section 4 is used to acquire the images with good quality, so these fixed thresholds can work well.

So far, we can detect a standard FIFA ball globally without color classification using our omnidirectional vision system.

5.3. Ball tracking by integrating the ball speed estimating algorithm

In the competition, there is no need to conduct the global detection above by processing the whole image in every frame. Once the ball has been detected globally, we can track the ball by integrating

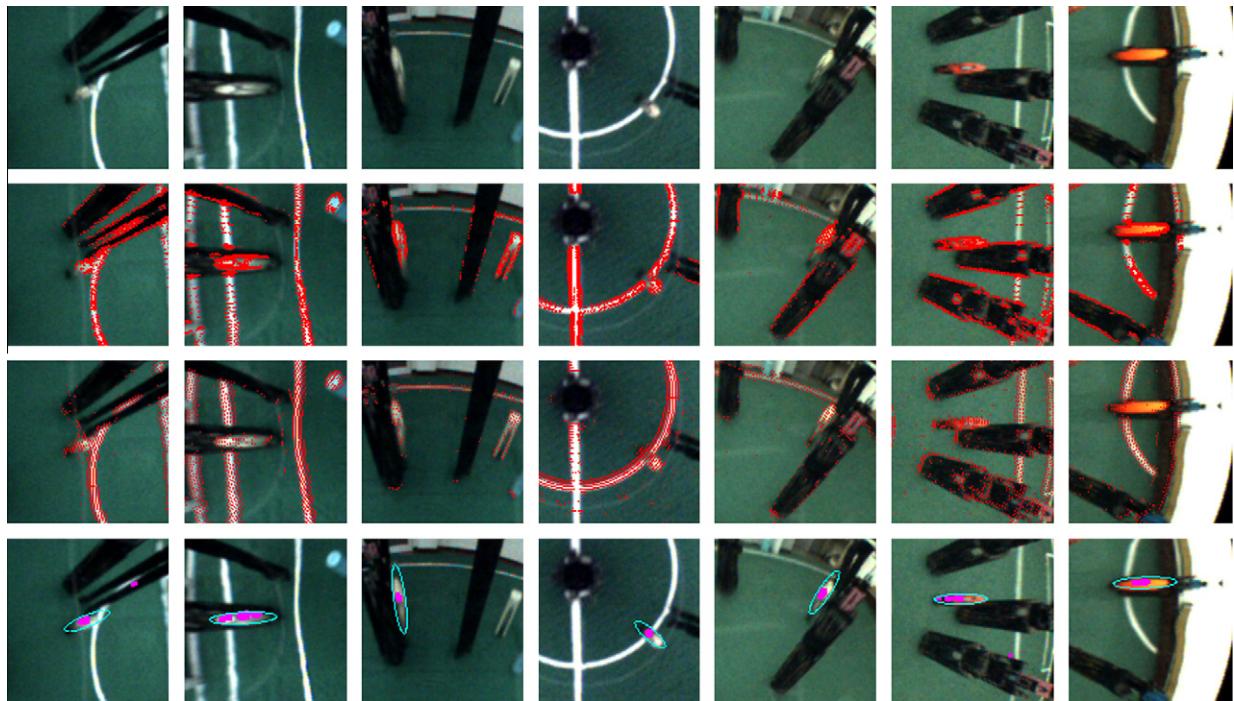


Fig. 21. Some of the global detection results in doing the performance statistics for our ball recognition algorithm. (First row) The panoramic images. (Second row) The possible contour points detected by rotary scan. (Third row) The possible contour points detected by radial scan. (Fourth row) The ball recognition results.

a ball speed estimating algorithm proposed by Lauer et al. [41]. The algorithm models the speed estimation as a standard linear regression problem, and uses ridge regression to solve it. In the actual application, ball tracking is started only when the same ball has been detected globally in three consecutive frames, for at least three frames of ball information are needed for estimating the ball speed reliably. Also, some false positives caused by image noise can be eliminated from being tracked using this method, making the ball tracking more reliable. During the tracking process, the estimated ball speed is used to predict the ball position of the next frame in the real world coordinate and the ellipse position imaged by the predicted ball can be calculated according to the calibration result of the omnidirectional vision. So we only need to process the nearby image region of the predicted ball position with the same image processing algorithm, and the running time needed can be reduced greatly. The nearby image region changes dynamically with the major-axis and the minor-axis of the ellipse imaged by the predicted ball. When the ball has been lost for several consecutive frames during the tracking process, the global detection and the ball speed estimating algorithm should be restarted.

Furthermore, in order to improve the correct detection rate, we would consider that a candidate ellipse exists, which is corresponding to each of the possible ellipse center points after rotary scan and radial scan without checking the coinciding points, and then search for the ball in more candidate ellipses during the tracking process. So when the ball is occluded partially but less than half, it can still be detected.

5.4. The experimental results

First, we demonstrate the process and the results of our ball recognition algorithm by processing the typical panoramic image. The panoramic image, the results of the rotary scan and radial scan are shown in Fig. 20a–c respectively. The red points in Fig. 20b and c are all of the possible contour points. The final recognition result is shown in Fig. 20d, in which the purple points are the centers

of the candidate ellipses, the yellow rectangles are tangent to the candidate ellipses, and the cyan ellipse is the theoretical imaging of the detected ball in the panoramic image. From Fig. 20, we see that although lots of other objects exist around the robot and the image background is complex, the FIFA ball can be detected successfully without any color classification.

With the aim to acquire the statistics of the correct detection rate and the false positives of our recognition algorithm, we collected 137 different panoramic images with standard FIFA balls in the cluttered environment. These balls were different in color and texture. They also were not occluded in the images. The distance between these balls and the robot itself were restricted to be less than 4.5 m, for the ball would be imaged very small if it was too far from the robot itself. After processing these images one by one, 132 balls were globally detected, and the false positives was 2. So the correct detection rate was 96.35%. Only global detection was dealt with in this statistics. The correct detection rate can be increased further by combining object tracking algorithms. Therefore the correct detection rate and the false positives are acceptable for soccer robots to play competitions with an arbitrary FIFA ball. Some of the recognition results are demonstrated in Fig. 21.

We also tested the ball tracking algorithm proposed in Section 5.3. Several results of recognizing and tracking the ball in a test sequence of panoramic images are shown in Fig. 22. Fig. 22a shows the result of global recognition, and Fig. 22b–n the results of the tracking process. The blue ellipses are the predicted ball positions, and the cyan ellipses are the real detected balls. From Fig. 22, we see that our algorithm tracked the ball efficiently. Even when the ball was temporarily fully occluded as shown in Fig. 22c, the tracking algorithm could redetect the ball by using the ball speed that was estimated in the past frames to predict the ball position in the next frame. Because more candidate ellipses are considered without checking the coinciding points, the balls occluded less than half can also be detected successfully as shown in Fig. 22j and m.

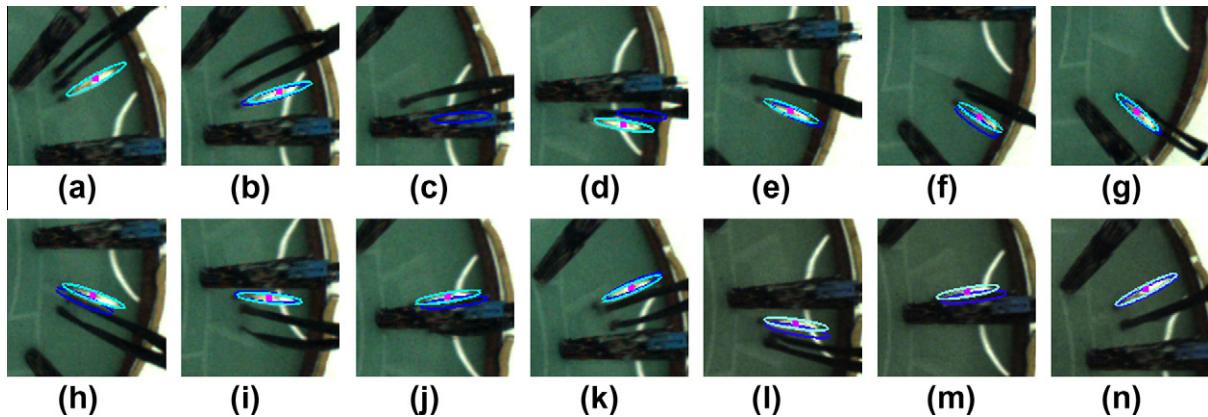


Fig. 22. The results of our ball tracking algorithm. (a) The global detection result. (b–n) The ball tracking results.

The RoboCup MSL competition is highly dynamic and the robot must process its sensor information as soon as possible. We tested the running time of our recognition algorithm. The robot's computer is equipped with a 1.73G CPU and 512M memory. It takes about 100–150 ms to realize global detection by analyzing a whole panoramic image with the dimension of 444×442 . However, once the ball has been detected globally, the running time can be reduced to be 4–20 ms in the tracking process for only the partial image near the predicted ball position is needed to be processed. Therefore our recognition algorithm meets the real-time requirement of the RoboCup MSL competition.

5.5. Discussion

When compared to other existing ball recognition methods without color information, our algorithm has the following advantages:

- It does not need any learning or training process which is necessary in the recognition algorithm based on Adaboost learning.
- It can deal with global detection which is not considered in the CCD algorithm.
- It is based on the omnidirectional vision system, so it can detect the ball more quickly in a much larger range than other existing methods that only use perspective cameras.
- It can incorporate object tracking algorithms easily to detect the arbitrary ball more efficiently and in real-time. Its interim and final results can also be used as important clues for other recognition methods.
- The idea of our algorithm based on our omnidirectional vision can be used in any other omnidirectional or perspective vision systems, if the imaging characteristic of the ball can be analyzed in advance.

However, there are still some deficiencies in our algorithm. The first deficiency is that the imaging of the ball is occluded partly by the robot itself when the ball is too close to the robot according to the imaging characteristic of our omnidirectional vision system [42], so our algorithm fails in recognizing this ball globally. This deficiency can be solved by adding a perspective camera to recognize the arbitrary ball as we had demonstrated successfully in the second technical challenge of RoboCup2007 Atlanta. In the algorithm, we apply the Sobel filter to detect all the edge points and their gradient directions in the perspective image first, and then recognize the arbitrary ball by using Hough transform based on the gradient information to detect the circle imaged by the ball. By using the omnidirectional vision and perspective camera to-

gether, the robot can recognize the ball globally, track the ball and perform precise ball manipulation such as dribbling and shooting. The video of our robot's tracking and playing with an arbitrary FIFA ball can be found on our team website: <http://nubot.nudt.edu.cn/2009videoen.htm>.

The second deficiency is that the ball can be recognized effectively only when its distance to the robot is less than 4.5 m, which is not sufficient for the RoboCup MSL with large field of $18 \text{ m} \times 12 \text{ m}$. This deficiency can be solved by planning paths for robots to search for the ball on the field, as shown in our video mentioned above. Multi-robot cooperative sensing could also be used to deal with the problem, and every robot could share the ball information with its teammates. The third deficiency is that our algorithm can only deal with the situation in which the ball is on the ground. We have to develop some arbitrary ball recognition method based on a stereo-vision system to solve this problem. The fourth deficiency is that when the ball is occluded or even partially occluded for a long time (more than several frames), our algorithm fails to track it. This deficiency may be solved by incorporating other tracking algorithms or other recognition methods such as Adaboost learning algorithm.

6. Conclusion and future work

In this paper, we aim to develop a robust omnidirectional vision sensor for the RoboCup MSL soccer robots to be able to work well under varying lighting conditions and without the constraint of the current color-coded environment. In the image acquisition, we propose a novel method to auto-adjust the camera parameters based on image entropy. Image entropy is verified by experiments to be able to indicate whether camera parameters are properly set, so camera parameters can be auto-adjusted based on it to achieve some kind of color constancy for the output of vision systems, which is validated by the experimental results of color object recognition and the robot's self-localization under different lighting conditions. In the image processing, we design a novel algorithm for ball recognition without color classification based on our omnidirectional vision system. The conclusion is derived that the ball on the field can be imaged to be an ellipse approximately in panoramic images, so the ball can be recognized without color classification by detecting the ellipse with our own image processing algorithm. The experimental results show that the arbitrary FIFA ball can be recognized and tracked effectively in real-time by our method, even if the environments are cluttered.

However, according to the discussions in Sections 4 and 5, there are still lots of work needed to be done to achieve complete robustness for the omnidirectional vision under dynamic lighting condi-

tions and in generic, but not color-coded, soccer competition environment. In camera parameters auto-adjustment, object recognition and tracking algorithm should be integrated into our method. So when the illumination is highly nonuniform, the robot can adjust its camera parameters based on the image entropy of the object region. The color constancy of the object region can be achieved, which is similar to the visual attention mechanism. When non-omnidirectional vision system is used, the robot can also use the object region as referenced information to judge whether camera parameters should be adjusted. In the arbitrary FIFA ball recognition, more effective tracking algorithms or other recognition methods should be integrated into our algorithm, so the robot can recognize and track the ball more effectively even when the ball is occluded frequently. Furthermore, recognizing the arbitrary FIFA ball in three dimension space should be researched, because the ball is often lifted by the robots' high kicks during the current RoboCup MSL competition.

Acknowledgement

The authors would like to thank the anonymous reviewers for their valuable comments.

References

- [1] Takahashi Y, Nowak W, Wisspeintner T. Adaptive recognition of color-coded objects in indoor and outdoor environments. RoboCup 2007: Robot Soccer World Cup 2008;XI:65–76.
- [2] Bonarini A, Aliverti P, Lucioni M. An omnidirectional vision sensor for fast tracking for mobile robots. *IEEE Trans Instrum Meas* 2000;49(3):509–12.
- [3] Menegatti E, Pretto A, Scarpa A, Pagello E. Omnidirectional vision scan matching for robot localization in dynamic environments. *IEEE Trans Robotics* 2006;22(3):523–35.
- [4] Lauer M, Lange S, Riedmiller M. Calculating the perfect match: an efficient and accurate approach for robot self-localization. RoboCup 2005: Robot Soccer World Cup 2006;IX:142–53.
- [5] Mayer G, Utz H, Kraetzschmar GK. Playing robot soccer under natural light: a case study. RoboCup 2003: Robot Soccer World Cup 2004;VII:238–49.
- [6] Ristić D, Vuppala SK, Gräser A. Feedback control for improvement of image processing: an application of recognition of characters on metallic surfaces. In: Proceedings of the 4th IEEE international conference on computer vision systems; 2006.
- [7] Grillo E, Matteucci M, Sorrenti DG. Getting the most from your color camera in a color-coded world. RoboCup 2004: Robot Soccer World Cup 2005;VIII:221–35.
- [8] Lunenburg JJM, Ven Gvd, Tech united team description. In: Proceedings of the RoboCup 2008 Suzhou, CD-ROM; 2008.
- [9] Neves AJR, Cunha B, Pinho AJ, Pinheiro I. Autonomous configuration of parameters in robotic digital cameras, in: Proceedings of the 4th Iberian conference on pattern recognition and image analysis, ibPRIA 2009; 2009.
- [10] Agarwal V, Abidi BR, Koschan A, Abidi MA. An overview of color constancy algorithms. *J Pattern Recogn Res* 2006;1(1):42–54.
- [11] Forsyth DA. A novel algorithm for color constancy. *Int J Comput Vision* 1990;5:5–36.
- [12] Mayer G, Utz H, Kraetzschmar GK. Towards autonomous vision self-calibration for soccer robots. In: Proceedings of the international conference on intelligent robots and systems; 2002. p. 214–9.
- [13] Gönner C, Rous M, Kraiss K. Real-time adaptive colour segmentation for the RoboCup middle size league. RoboCup: 2004 Robot Soccer World Cup 2005;VIII:402–9.
- [14] Lu H, Zheng Z, Liu F, Wang X. A robust object recognition method for soccer robots. In: Proceedings of the 7th world congress on intelligent control and automation; 2008. p. 1645–50.
- [15] Anzani F, Bosisio D, Matteucci M, Sorrenti DG. On-line color calibration in non-stationary environments. RoboCup 2005: Robot Soccer World Cup 2006;IX:396–407.
- [16] Heinemann P, Sehnke F, Streichert F, Zell A. Towards a calibration-free robot: the ACT algorithm for automatic online color training. RoboCup 2006: Robot Soccer World Cup 2007;X:363–70.
- [17] Hanek R, Schmitt T, Buck S, Beetz M. Towards RoboCup without color labeling. RoboCup 2002: Robot Soccer World Cup 2003;VI:179–94.
- [18] Treptow A, Zell A. Real-time object tracking for soccer-robots without color information. *Robot Auton Syst* 2004;48:41–8.
- [19] Kunii T, Sugiura H, Matoba N. A new automatic exposure system for digital still cameras. *IEEE Trans Consum Electr* 1998;44(1):192–9.
- [20] Chikane V, Fuh C. Automatic white balance for digital still cameras. *J Inf Sci Eng* 2006;22(3):497–509.
- [21] Chern NNK, Neow PA, Ang Jr MH. Practical issues in pixel-based autofocusing for machine vision. In: Proceedings of the 2001 IEEE international conference on robotics and automation; 2001. p. 2791–6.
- [22] Gooßen A, Rosenstiel M, Schulz S, Grigat R. Auto exposure control for multi-slope cameras. In: Proceedings of ICIAR 2008; 2008. p. 305–14.
- [23] Hanek R, Beetz M. The contracting curve density algorithm: fitting parametric curve models to images using local self-adapting separation criteria. *Int J Comput Vision* 2004;59(3):233–58.
- [24] Hanek R, Schmitt T, Buck S, Beetz M. Fast image-based object localization in natural scenes. In: Proceedings of the 2002 IEEE/RSJ international conference on intelligent robots and systems; 2002. p. 116–22.
- [25] Mitri S, Pervölk K, Surmann H, Nüchter A. Fast color-independent ball detection for mobile robots. In: Proceedings of IEEE mechatronics and robotics, Aachen, Germany; 2004. p. 900–5.
- [26] Mitri S, Frintrop S, Pervölk K, Surmann H, Nüchter A. Robust object detection at regions of interest with an application in ball recognition. In: Proceedings of IEEE international conference on robotics and automation; 2005. p. 125–30.
- [27] Coath G, Musumeci P. Adaptive arc fitting for ball detection in RoboCup. In: Proceedings of the APRS workshop on digital image analysing; 2003.
- [28] Bonarini A, Furiani A, et al. Milan RoboCup team 2009. In: Proceedings of the RoboCup 2009 Graz, CD-ROM; 2009.
- [29] Neves AJR, Azevedo JL, et al. CAMBADA'2009: Team Description Paper. In: Proceedings of the RoboCup 2009 Graz, CD-ROM; 2009.
- [30] Martins DA, Neves AJR, Pinho AJ. Real-time generic ball recognition in RoboCup domain. In: Proceedings of the 11th edition of the Ibero-American conference on artificial intelligence, IBERAMIA 2008, Lisbon, Portugal; 2008.
- [31] Zweigle O, Käppeler U-P, et al. 1. RFC stuttgart team description 2009. In: Proceedings of the RoboCup 2009 Graz, CD-ROM; 2009.
- [32] Zhang H, Lu H, Ji X, et al. NuBot team description paper 2007. In: Proceedings of the RoboCup 2007 Atlanta, CD-ROM; 2007.
- [33] Benosman R, Kang SB, editors. Panoramic vision: sensors, theory and applications. Springer-Verlag; 2001.
- [34] Voigtländer A, Lange S, Lauer M, Riedmiller M. Real-time 3 D ball recognition using perspective and catadioptric cameras. In: Proceedings of 2007 European conference on mobile robots; 2007.
- [35] Goshtasby AA. Fusion of multi-exposure images. *Image Vision Comput* 2005;23:611–8.
- [36] Shannon CE, Weaver W. The mathematical theory of communication. Urbana: University of Illinois Press; 1963/1949.
- [37] Gonzalez RC, Woods RE. Digital image processing. 2nd ed. Prentice Hall; 2002.
- [38] Marchant JA. Testing a measure of image quality for acquisition control. *Image Vision Comput* 2002;20:449–58.
- [39] Liu F, Lu H, Zheng Z. A modified color look-up table segmentation method for robot soccer. In: Proceedings of the 4th IEEE LARS/COMRob 07; 2007.
- [40] Zhang H, Lu H, Wang X, et al. NuBot team description paper 2008. In: Proceedings of the RoboCup 2008 Suzhou, CD-ROM; 2008.
- [41] Lauer M, Lange S, Riedmiller M. Modeling moving objects in a dynamically changing robot application. KI 2005: Advances in Artificial Intelligence 2005:291–303.
- [42] Lu H, Zhang H, Xiao J, Liu F, Zheng Z. Arbitrary ball recognition based on omnidirectional vision for soccer robots. RoboCup 2008: Robot Soccer World Cup 2009;XI:133–44.