Precise Extraction of Partially Occluded Objects by Using HLAC Features and SVM

Kazutoki Otake, Kazuhito Murakami, and Tadashi Naruse

Graduate School of Information Science and Technology, Aichi Prefectural University
Kumabari, Nagakute-cho, Aichi 480-1198 Japan
im071006@cis.aichi-pu.ac.jp,
{murakami,naruse}@ist.aichi-pu.ac.jp

Abstract. In the RoboCup competition, robot soccer game, ball and robots are extracted by using color information. If color markers attached on the robot or a ball itself are occluded, especially the occlusion ratio is high, it will be difficult to extract them. This paper proposes a new and high precision method which extracts partially occluded objects based on the statistical features of the pixel and its neighborhoods. Concretely, at first, input image is labeled by using color information and small candidate regions which have similar color to the color markers or the ball are extracted, then each candidate region is classified into partially occluded object or noise by using HLAC features and SVM. We applied our method to the global vision system of RoboCup small size league (SSL) and confirmed that it could extract partially occluded objects, 94.23% for 5 to 8 pixels area and 80.06% for 3 to 4 pixels area, and worked more than 60fps.

1 Introduction

RoboCup, robot soccer game, is an international research project to clarify and promote robot engineering and artificial intelligence by using autonomous robots and started on 1997. The final objective is “By the year 2050, develop a team of fully autonomous humanoid robots that can win against the human world soccer champion team.”

To extract robots and a ball, the images of the CCD cameras that are mounted at the ceiling of the hall or on the robot itself are used in general. The former is called “global vision” and the latter is called “local vision”, respectively. Figure 1 shows examples of the soccer scene in RoboCup competition. Figure 1(a) shows SSL’s robots which utilize a global vision system, and Fig.1(b) shows 4-legged league’s robot which utilizes a local vision system.

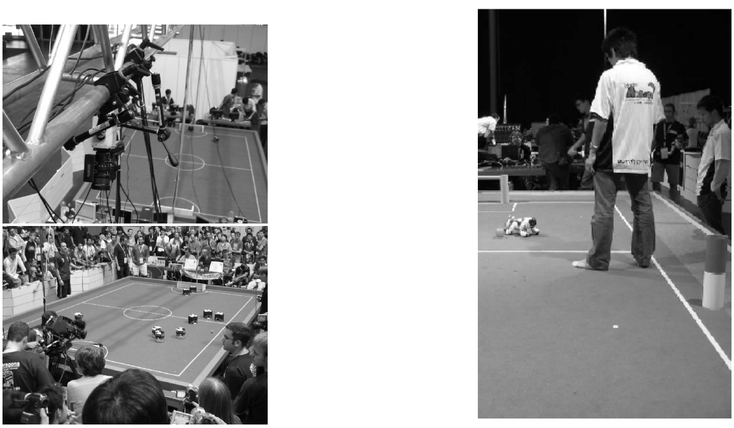
In the real game, many teams use color markers to extract and classify each robot and ball. There is no restriction, except for the team color marker, to the color and the shape of the color markers, so each team can uses different color/shape markers[1].

U. Visser et al. (Eds.): RoboCup 2007, LNAI 5001, pp. 17–28, 2008.

Figure 2 shows an example image obtained through a CCD camera. Figure 2(a) is an original image and Fig.2(b) is its labeled result. In the small size league, it needs less than 1/60 seconds to process 1 cycle. When all of the color markers are observed completely, the robots and ball can be detected, but if the objects are partially occluded, then it will be difficult to extract. This kind of occlusion occurs in the global vision system and also in the local vision system. Most of the conventional methods which calculate width, height, area, etc. of the candidate objects or regions have not been succeeded in extracting partially occluded objects and in judging noise or target object. As described above, it is necessary for the camera to observe whole of the object without occlusion. Narita et al. have reported that multi-cameras could reduce the occlusion of the ball for RoboCup SSL[2]. However, even though the number of cameras increases, the occlusion would not be 0.

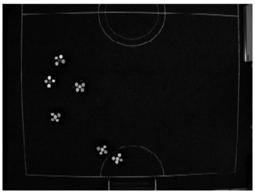
In general, particle filter or Kalman filter etc. are used for the object tracking with occlusion. Sugimura et al. have reported the robust tracking method for the object detection[3]. However, an essential problem remains that the moment when the object just begins to be seen could not be caught by the conventional tracking methods.

This paper proposes a new and high precision method to extract partially occluded objects. The candidate regions that are extracted and labeled by color information are classified into the target object or noise by using higher order local autocorrelation (HLAC) features and support vector machines (SVM). In the following parts, the analysis of extraction errors from the view point of extraction and occlusion are described in section 2, a high speed and robust method is explained in section 3, and the experimental results for the RoboCup vision system are expressed in sections 4 and 5.

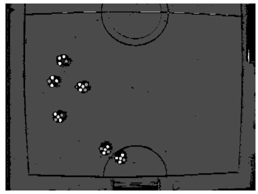


(a) Small-size robot (b) 4-legged robot

Fig. 1. Soccer scenes in RoboCup



(a) Original image



(b) Labeled result

Fig. 2. An image obtained through a global vision

2 Problems of Conventional Methods

2.1 Extraction Error by Noise

Noise appears due to the aberration of camera's lens or the threshold setting of the parameters such as the width, height, area, and so on, but when the noise has the same values as the target pixel, it will be difficult to remove it.

Usually, since the target is not so small, in general it is larger than the noise, conventional method succeeds in thresholding. If the extracted area becomes small by the occlusion, then the extraction rate goes down.

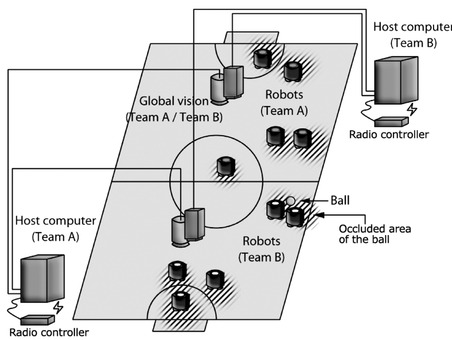


Fig. 3. Occluded area in SSL (Occlusion area is displayed with diagonal lines)

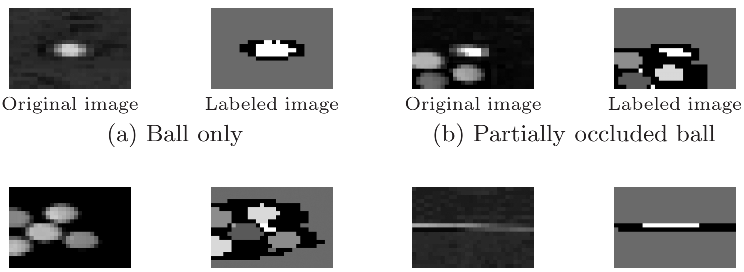
2.2 Extraction Error by Partial Occlusion

Many teams in SSL including our team use a global vision system composed of 2 or more cameras. Occlusion occurs a little in the center of the image, and on the contrary, it becomes more in the area far from the center of the image as shown in Fig.3.

Figure 4 shows a typical situation which causes occlusion. In the RoboCup SSL competition, a golf ball colored in orange is used. Figure 5 shows examples of labeled result for orange regions. Figure 5(a) is a case that only a ball exists and Fig.5(b) is a case that some part of a ball is occluded by a robot.

Figures 5(c) and 5(d) are the cases that some pixels around color marker or a line on the field are detected as pseudo orange region. In the RoboCup competition, it is necessary to classify these similar regions in real time whether it is a real target object or a noise.

Fig. 4. A typical scene of occlusion



Original image Labeled image Original image Labeled image

(a) Ball only (b) Partially occluded ball

Original image Labeled image Original image Labeled image

(c) Noise on the robot (d) Noise around the line

Fig. 5. Examples of orange objects

3 A Method to Extract Partially Occluded Object by HLAC and SVM

3.1 Principle of the Proposed Method

This section describes a method to extract partially occluded object by using higher order local autocorrelations (HLAC)[4] and support vector machines (SVM). Figure 6 shows the concept of the proposed method. First, the system calculates HLAC features by using 35 masks shown in Fig.7 for each pixel which belong to class 1 (i.e. ball) and class2 (i.e. noise), respectively. The number of dimension of HLAC features is $353 = 105$ because each pixel's value is composed of Y, U and V values. Then, the distributions of HLAC features for class1 and

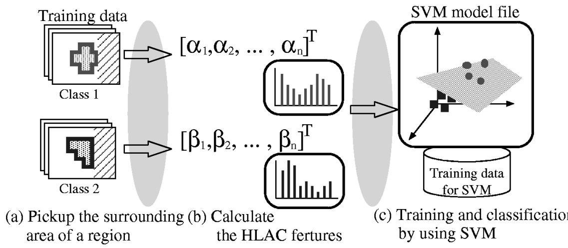


Fig. 6. The flow of the proposed method

class2 are applied to SVM. The combination of HLAC and SVM realizes a robust extraction to the changes of direction (occlusion occurs in any directions) and also to the changes of lighting.

3.2 Calculation of HLAC Features and Normalization

How to expand autocorrelation function to higher orders is presented, for example, in [3]. Let the reference value be r on the image f, then the N-th order autocorrelation $c(a_1, \cdots a_N)$ is defined by the calculation as

$$c(a_1, \cdots a_N) = \int f(r)f(r+a_1)\cdots f(r+a_N)dr, \tag{1}$$

here $a_1, a_2, ...a_N$ are the reference values around r.

Now, restrict the order N up to $2(N = 0, 1, 2)$ and its displacement is the correlation around the $f(r)$ at a maximum 3 in the 3×3pixels. Then the number of feature values becomes 35 as shown in Fig.7.

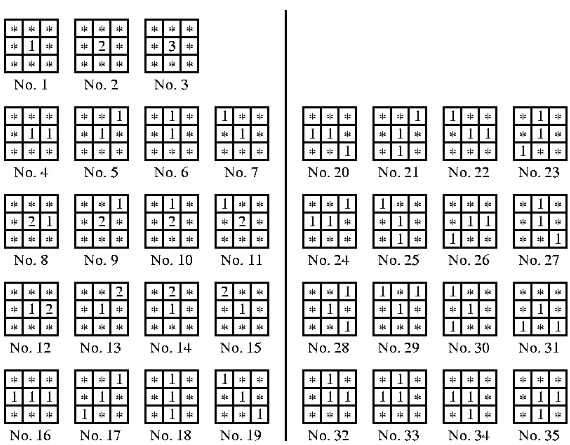


Fig. 7. Templates of higher order local autocorrelation feature that N is limited to 2

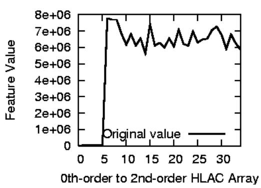
As for the feature vector calculated by the sum of products of pixel value, it is necessary to normalize so that the value of each feature vector be the same range. In case of 8-bit quantization for each pixel value, the normalized autocorrelation $c'(a_1, \cdots a_N)$ becomes as follows.

$$c'(a_1, \cdots a_N) = \sum_r \frac{f(r)f(r+a_1)\cdots f(r+a_N)}{255^N} \tag{2}$$

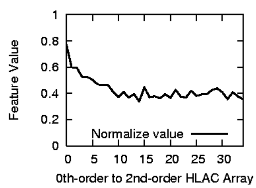
In addition, feature vectors are separately calculated for each element of the YUV color information, so the feature vector H is expressed as

$$H = \{c'_Y(a_1, \cdots a_N), c'_U(a_1, \cdots a_N), c'_V(a_1, \cdots a_N)\} \tag{3}$$

An example of H is shown in Fig.8. As shown in Fig.8(a), the feature values corresponding to $N = 0, 1(0$ to 2 for x-axes) are extremely smaller than others. In contrast, Fig.8(b) shows the normalized values.



(a) Before normalization



(b) After normalization

Fig. 8. An example of HLAC features

3.3 Training by SVM

It is reported that SVM is effective to the changes of the position and the luminance in the object extraction[5,6]. Feature vectors, H_{ball} for the object to be detected as the target object and H_{noise} for the noise, are used for the training data of SVM. These data are collected as the training data in the beginning and creates a model for SVM[1].

4 Application to RoboCup SSL

4.1 Time Restriction in RoboCup

The flow of the ball detection is shown in Fig.9.

First, labeling and segmentation[2] processes are applied to the color space. The proposed method uses the parameters such as width, height and area features like a conventional method.

[1] C-SVC is used for the classifier and RBF kernel is used in LibSVM[7].

[2] Segmentation process by using CMVision2[9].

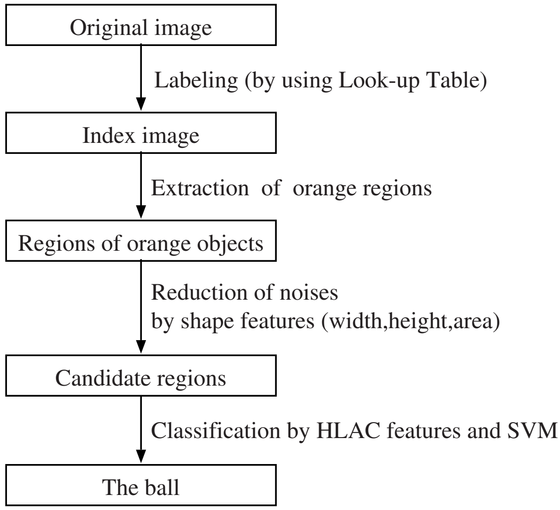


Fig. 9. Ball detection flow

In general, the processing interval is 60fps or more. And the total processing time for each frame is 16.7msec or less. In our conventional method, it took 1.9msec for image processing and 2.1 to 6.1msec for other processing such as strategy, pass generation, and so on. Therefore, our proposed method can use about $16.7 - 6.1 = 10.6$msec or less[8].

4.2 Host Computer System

The specification of our computer is as follows:

– CPU Athron 64 3500+
– 512MB Memory
– Debian Linux Operating system
– GV-VCP/PCI capture boards
(Bt848 chipset is popular and low-cost board.)

4.3 Characteristics of Robots and Ball in Occlusion

We have used the black color for the robot's body except for the color marker as shown in Fig.10. The height of the robot is 150mm and the ball is 42mm. When the ball is occluded, the color of the neighborhood region of the ball becomes black.

5 Experiment

5.1 Construction of Training Model and Classification

We have prepared training data with occlusion which occurred around periphery of the image, and at the same time, under the condition that the lighting changes in

Fig. 10. A robot and a ball that are used for occlusion experiment

order to rise up the robustness of the proposed method. The occlusion images are obtained by controlling a robot to occlude some part of the ball as shown in Fig.10.

We collected the features of a ball and noise from orange region of size 8 3 pixel, because the size of the ball was observed more than 83 pixel without occlusion under our experimental conditions. The collected training data are shown in Table 1. The distributions of the training data are shown in Fig.11. Figures 11(a), 11(b) and 11(c) are the distributions of the width, height and area of the ball, respectively, and Fig.s11(d), 11(e) and 11(f) are those of the noise.

Table 1. The number of training data of ball and noise from orange region

Picked up region		The number of training data for balls	The number of training data for noise
Width×Height [pixel]	Area [pixel]		
8x3 or less	1	300	2000
8x3 or less	2	1500	2000
8x3 or less	3,4	1800	2000
8x3 or less	5-8	2000	2000
8x3 or less	9-16	2000	0
8x3 or less	17-24	2000	0

The classifier based on SVM was created by these training data. We evaluated the classifier by using the 10-fold cross validation. From Fig.11, all of noise observed became 8 pixel or less, then the experiment and classification for partially occluded balls were done for under 8 pixels.

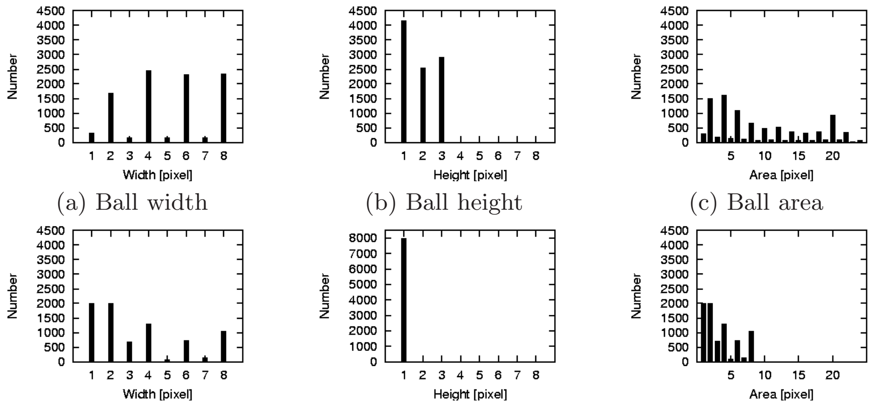
The training data was evaluated by using the following expressions.

$$precision = \frac{R}{N} \tag{4}$$

$$recall = \frac{R}{C} \tag{5}$$

$$F - measure = \frac{2 \cdot precision \cdot recall}{precision + recall} \tag{6}$$

here, R is the the number of orange regions of the detected balls (noise is not included), N is the the number of orange regions of the detected balls (noise is included), and C is the the number of orange regions of the balls in training data set.



(a) Ball width (b) Ball height (c) Ball area

(d) Noise width (e) Noise height (f) Noise area

Fig. 11. Distributions of shape features of training data

The ball has been detected only by using width, height and area of the region in a conventional method. Therefore, not only the ball but also the noise is included in the region when the size becomes smaller. The detection rate D of the ball in a conventional method is calculated as $D = \frac{B}{B+N}$ for each area of the region, here, B and N are the numbers of balls and noise observed in the unit time.

The robot shown in Fig.10 has moved at random by using a remote controller, and the numbers of observations of the ball and noise were examined. Table 2 shows this result. The detection rate of a conventional method was calculated by using it. The number of frames in which the orange regions were observed was assumed to be 6000 frames.

5.2 Calculation of Detection Rate in a Practical System

In our practical system, in the unit time for ball detection rate $D^{(n)}$ is calculated from the product of the classification rate $BallClassificationRate(n)$ and the number $B^{(n)}$ of the ball observed by a certain number n of pixels.

$$D(n) = \frac{\sum_{i=n}^{24} \left(B^{(i)} \cdot BallClassificationRate(i) \right)}{\sum_{i=n}^{24} B^{(i)}} \tag{7}$$

Table 2. The numbers of orange regions for ball and noise observed in 6000 frames

Picked up region		The numbers of orange regions	
Width×Height [pixel]	Area [pixel]	ball	noise
8x3 or less	1	62	8879
8x3 or less	2	173	8114
8x3 or less	3,4	304	5388
8x3 or less	5-8	490	7109
8x3 or less	9-16	1518	0
8x3 or less	17-24	2086	0

From the data in Table 2 and classification result, ball's detection rate $D(i)$ ($i = 1, 2, 3, 5, 9$) was calculated and the processing time was measured.

6 Experimental Result

6.1 Classification Result by Each Area

Table 3 expresses the experimental result for the object classification of a conventional method and our proposed method for each area.

6.2 Detection Result in a Practical System

Table 4 shows the detection rate and the processing time in our practical system. The detection rate was calculated based on ball's samples in continuous observed data(6000 frames).

Although our proposed method can be applied to both of the ball detection process and the marker detection process, in the experiment, we applied only to the ball detection. The processing time shown in Table 4 denotes all of the vision processing.

7 Discussions

Table 3 shows the performance of the classification in each area. The proposed method is higher than a conventional method, and especially for the case that the size of the object is small. Furthermore, it is robust for the changes of lighting intensity, concretely it could work even if it changes in 1.1ev(+214%).

Table 4 shows the real detection rate in our particle system. RoboCup system requires more than 97 or 98% extraction rate. Although a conventional method has the limit that the target size should be more than 2 pixels area, and our proposed method has enough potential for all cases.

For the object of size 3 pixels area or more, it is known from Table 4 that the processing time is less than 10.6msec as discussed in 4.1, and it satisfies our system's condition to work in real time.

Table 3. Classification result for each area

Picked up region		Classification result		
Width × Height [pixel]	Area [pixel]	*precision*	*recall*	*F-measure*
8x3 or less	1	59.00%	45.12%	51.13%
8x3 or less	2	71.80%	73.88%	73.88%
8x3 or less	3,4	73.83%	87.43%	80.06%
8x3 or less	5-8	93.05%	95.45%	94.23%
8x3 or less	9 or more	100.00%	100.00%	100.00%

Table 4. Detection result and processing time in our practical system

Picked up region		Conventional method		Proposed method	
Width × Height [pixel]	Minimum area [pixel]	Detection rate	Processing time[ms]	Detection rate	Processing time[ms]
8x3 or less	1	78.94%	1.84	96.49%	120.31
8x3 or less	2	88.00%	1.83	97.09%	14.50
8x3 or less	3	83.03%	1.71	97.98%	9.86
8x3 or less	5	88.00%	1.69	99.31%	9.56
8x3 or less	9	100.00%	1.65	100.00%	8.53

8 Conclusions

This paper described a high speed method to extract partially occluded object by using HLAC and SVM. And it is shown that this method is robust for the changes of luminance. From experimental result, it has clarified to have the effectiveness for the extraction of partially occluded objects and also confirmed that it works in real time. We applied our proposed method to the RoboCup's global vision system and confirmed that it could extract color markers on the robots and a ball in the occluded situations.

Although this method works in real time, there still remains some subjects to be solved. It is required to realize more high speed processing in order to use the rest time for the strategic motion planning, and also it is also necessary to select more effective features to shorten the training process. These are coming subjects.

References

1. Bruce, J., Veloso, M.: Fast and Accurate Vision-Based Pattern Detection and Identification. In: Proc. of ICRA 2003, the 2003 IEEE International Conference on Robotics and Automation, Taiwan (May 2003)
2. Narita, R., Murakami, K., Naruse, T.: A Better Solution of Multi-camera's Layout for RoboCup Small Size League. In: Proc. of the 13th Japan-Korea Joint Workshop on Frontiers of Computer Vision, 2007(FCV 2007), pp. 351–356 (2007)
3. Sugiura, D., Kobayashi, Y., Sato, Y., Sugimoto, A.: People Tracking with Adaptive Environmental Attributes using the History of Human Activity, IPSJ SIG Notes. In: CVIM, vol. 115, pp. 171–178 (November 2006) (in Japanese)
4. Otsu, N., Kurita, T.: A new scheme for practical flexible and intelligent vision systems. In: Proc. of IAPR Workshop on Computer Vision, pp. 431–435 (1988)
5. Pontil, M., Verri, A.: Support vector machines for 3D object recognition. Proc. of IEEE Trans. PAMI 20(6), 637–646 (1998)
6. Okabe, T., Sato, Y.: Support Vector Machines for Object Recognition under Varying Illumination. In: Proc. of IPSJ Transctions on Computer Vision and Image Media, vol. 44, pp. 22–29 (2003) (in Japanese)

7. LIBSVM – A Library for Support Vector Machines, http://www.csie.ntu.edu.tw/~cjlin/libsvm/
8. Otake, K., Murakami, K., Naruse, T.: A High Speed and Robust Method to Extract Partially Occluded Objects for RoboCup Small Size League. In: Proc. of the 13th Japan-Korea Joint Workshop on Frontiers of Computer Vision, 2007(FCV 2007), pp. 63–68 (2007)
9. CMVision, http://www-2.cs.cmu.edu/~jbruce/cmvision/