Shape Based Round Object Detection Using Edge Orientation Histogram

Hamid Mobalegh, Lovísa Irpa Helgadóttir, and Raúl Rojas

Institut für Informatik, Freie Universität Berlin

Abstract. In this paper we introduce a shape based method to globally detect the ball in a RoboCup soccer scenario. The method can be used for any round object with detectable edges. The concept of integral images presented in Viola & Jones 2001, is used, however the integration is applied to a vector representation of the gradient orientation histogram of each pixel. The method takes advantage from the fact that large areas of the image can be filtered out, as these are only covered by straight edges. An overlapped binary search quickly reduces the search area and locates ball candidates in the image. The candidates are finally selected using an outlier elimination technique.

1 Introduction

Shape based object detection methods are often more reliable than methods that rely on color information. However, this is generally achieved at the cost of more processing power. The computational effort can be strongly reduced if the generality of the scenario is limited.

RoboCup Soccer is such an example. According to the RoboCup 2050 mission, robots shall be capable of tolerating outdoor lighting conditions and play with a standard FIFA ball. To reach this goal, the rules are shifted year by year to motive researches.

The work presented here is inspired by the previous work on *Histogram of oriented Gradients*[1] introduced by Dalal and Triggs[4]. Originally, the HOG algorithm focused on the problem of pedestrian detection in static images. Today it has expanded to other objects such as animals, vehicles and other media such as video streams. The method is of great importance as it is only based on the shape data of the image rather than other more environment-dependent information like brightness and color. The method we present here extends this idea by using integral images and an overlapped binary search. To accelerate the algorithm we were also inspired by intensity integral images presented in Viola et al. [12].

Ball detection without color information has been focused on in RoboCup competitions since its early years, as the RoboCup 2050 goal includes the use of a standard FIFA ball. The use of an arbitrary ball in RoboCup competitions is still limited to the "technical challenges" in many leagues. The biggest improvements

[1] HOG

S. Behnke et al. (Eds.): RoboCup 2013, LNAI 8371, pp. 326–335, 2014.

have occurred in the middle size league. A descriptive survey on the developments in this league is given in [6]. Researchers have different approaches to tackle this problem. Hough transformation is used by [9],[1] and [7]. It is computationally expensive and cannot be used for a global search. Arc and Model fitting are used in [5] and [3]. Both solutions are iterative and need a prior knowledge of the ball position. Partial occlusion and clutter is well tolerated by these methods. Another group of methods focuses on Adaboost classifiers (see e.g.[11] and [8]). These methods have been shown to have a low false positive rate.

Many of the algorithms addressed above fail to perform globally, due to their computational complexity. A local ball search is however more relevant in middle size and small size leagues as the camera does not have fast motion and the position of the ball can be predicted fairly well. In contrast, in the humanoid leagues the camera moves very fast and a prediction is not possible due to high amount of measurement error. Therefore more focus is needed on global search methods.

The rest of this paper is organized as follows: First we show how a gradient image is calculated. A description of how a histogram of the edge orientation is constructed follows. Then we explain how a intensity integral images can be used to accelerate our algorithm and represent the original image. Next we describe the overlapped binary searching algorithm that find the possible ball candidates and the statistical test we use to eliminate outliers. Finally we show the results of the algorithm tested on images recorded on a RoboCup humanoid field.

2 Structure of Method

The method is structured as presented in figure 1. It includes five stages starting with the calculation of the gradient vector. This is common in many shape based object detection techniques. Using non-maxima-suppression, the edges are thinned and normalized. A so-called "*Histogram Integral Image*"[2] is then constructed based on the orientation of the gradient vector in each pixel. This representation of the original image helps to accelerate the search algorithm, especially, as it is possible to implement this part in hardware. An overlapped binary search recursively scans the pyramid down and finds the best-fitting box around the object using edge orientation statistics from the HII. The final result is then once more filtered using further statistical criteria.

2.1 Gradient Vector Calculation and Thresholding

The gradient vector can be calculated using one of the standard methods. For a better performance we use Robert's cross operator[10]. Strong edges are selected using a simple thresholding. Non-maxima-suppression is then performed along the gradient direction similar to the method used in canny edge detector [2]. This reduces the edge thickness and provides normalized results for the procedure introduced in section 2.5. Figure 2 shows the intermediate results.

[2] HII

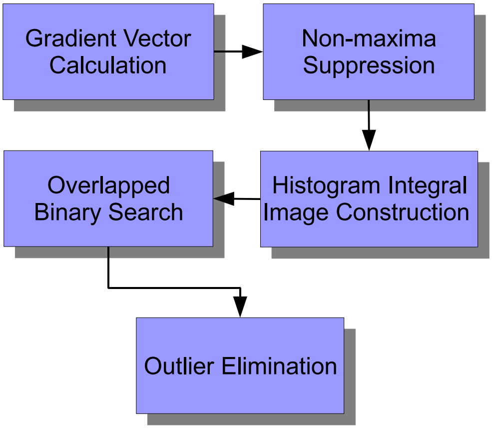
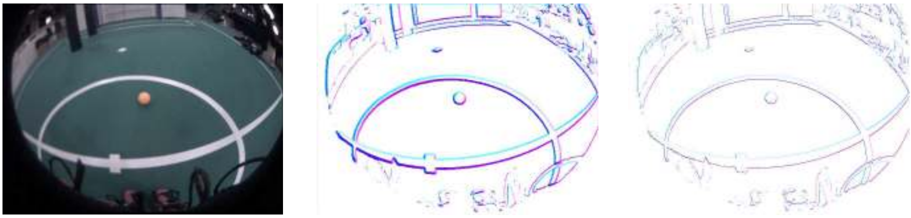


Fig. 1. Structure of the Ball Detection Method: First the gradient image is calculated. Then it is non-maxima-suppressed. The data is used to calculate the HII. An overlapped binary search locates the ball candidates. Finally outlier canidates are filtered based on statistical criteria.



a) Original Image b) Thresholded Edges c) Non-maxima-suppression

Fig. 2. Edge Points Selected for Ball Detection Algorithm: Gradient of the input image is calculated. Strong edges are selected upon thresholding. Edges are thinned using non-maxima suppression.

2.2 Histogram of Edge Orientations

A histogram of edge orientations is an important measure used in shape-based object detection algorithms. In our method, the histogram presents the distribution of edge directions. The direction of the gradient vector is extracted in a 360° range, which is quantized into 18 groups of 20 degrees each. The detection algorithm relies indirectly on the fact that an ideal round object has a uniform distribution of edge orientations. However, this feature gets lost rapidly as the window size grows and other contents are added to the image. Additional contents of the image always add positive values to the histogram.

A direct result gained from this is that large regions of the image can be entirely rejected if the orientation histogram contains zeros in more than a given number of directions. Theoretically, an edge orientation histogram belonging to a window containing the round object must be zero free. This, however, cannot always be fulfilled in real test conditions due to shadows, partial occlusions or some other effects. Therefore a certain number of zeros are allowed in the histogram, which can be adjusted in accordance to the image condition.

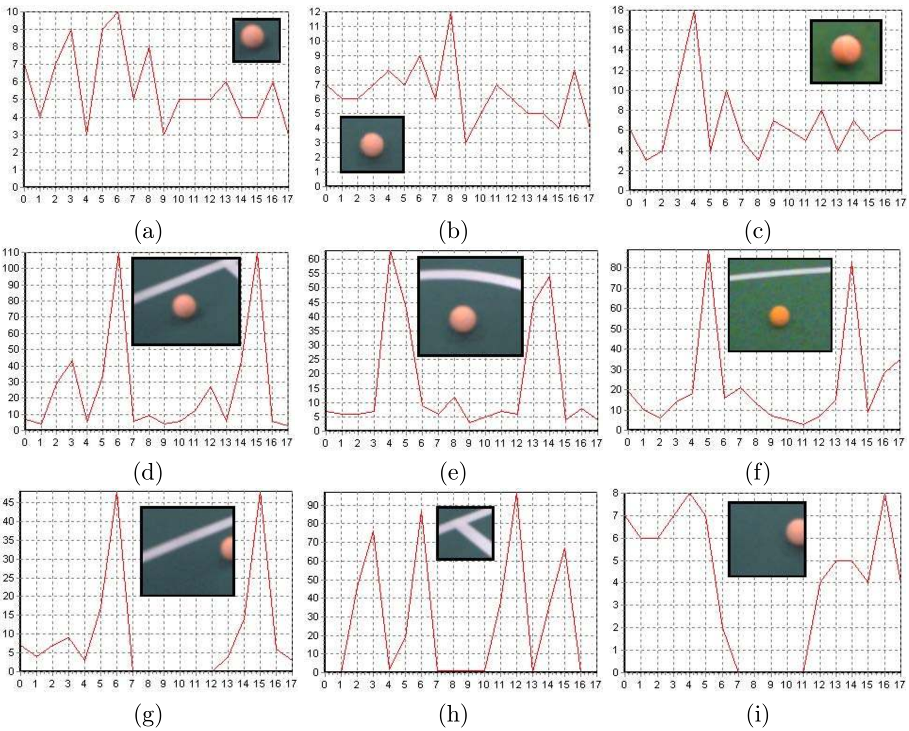


Fig. 3. Histogram of Edge Orientations: Examples of windows containing an entire ball (a-f), a partial view of the ball (g,i) or no ball (h). Only the examples without a complete ball (g-i) have bins containing zero pixels.

To visualize what kind of information an edge orientation histogram provides and how strong this measure is, several example histograms of the images captured from the robots are presented in figure 3. In figure 3a, b and c, the histograms of windows containing only a ball are presented. The histograms show more or less a uniform distribution over all directions. In figure 3c, d and e, the window size is enlarged and some other content is added. The histograms have lost the uniformness, however all components have remained non-zero. In figure 3g and i, the ball is partially visible which has lead to a significant number of zero components and finally the histogram of a ball-free window is shown in figure 3h. As it can be seen, a round object placed on the border of a window cannot be detected. A solution to this problem is discussed in section 2.4.

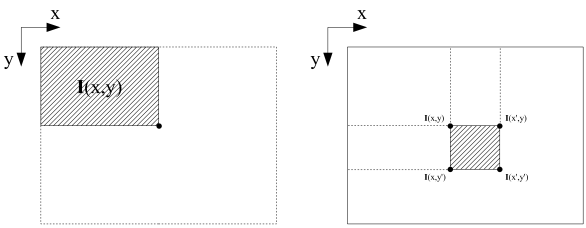
2.3 Histogram Integral Image

The idea of intensity integral images is introduced in [12]. In this work we extend this idea to the histogram of edge orientation.

Assume an image with 18 channels, i.e. each pixel value is an 18 dimensional vector. The vector describes the histogram of edge orientations according to a rectangle, stretched from the origin of the source image to the given coordinates as presented in figure 4a. This is described in equation 1.

$$\mathbf{I}(x, y) = \sum_{\substack{x\prime < x \\ y\prime < y}} \mathbf{H}(x\prime, y\prime) \tag{1}$$

$\mathbf{H}$ is a vector filled with zeros except for the component corresponding to the direction of the gradient vector at $(x\prime, y\prime)$, which is filled with 1 if the pixel is identified as an edge pixel.



(a) Calculation of HII (b) Computation of the Histogram Based on the Integral Representation

Fig. 4. Computation of Histogram Integral Image: Each entry in the HII is a vector representing the edge orientation histogram of a rectangle from the origin of the image to the indexed pixel. Based on this image a calculation of the histogram of an arbitrary window is simply done using 4 vector additions.

According to [12], an integral image can be computed in a single scan by using the following recurrence relation:

$$\mathbf{S}(x, y) = \mathbf{S}(x - 1, y) + \mathbf{H}(x, y) \tag{2}$$

$$\mathbf{I}(x, y) = \mathbf{I}(x, y - 1) + \mathbf{S}(x, y) \tag{3}$$

where $\mathbf{S}$ is a temporary vector holding a histogram of the current line of the image. It is enough to store $\mathbf{S}$ as a single accumulator because there is no reference to its history.

Integral representation reduces the computation of the histogram for any given window to two additions and one subtraction as follows:

$$\mathbf{Hist}(x, y, x\prime, y\prime) = \mathbf{I}(x, y) + \mathbf{I}(x\prime, y\prime) - (\mathbf{I}(x\prime, y) + \mathbf{I}(x, y\prime)) \tag{4}$$

This accelerates the operation to a great extent.

Algorithm 1. Overlapped Binary Search

```
Boolean search_ball(window, level)

begin
        if        window is already scanned then
                  return     true
        window    <- scanned
        calculate_histogram(window)
        if        histogram has at least one zero component then
                  return     false
        if        (level>5) then
                  return     false
        b         <- false
        for       all sub windows
                  b          <- b or search_ball(sub window, level+1)
        if        not b then
                  push_ball_candidate(window)
end
```

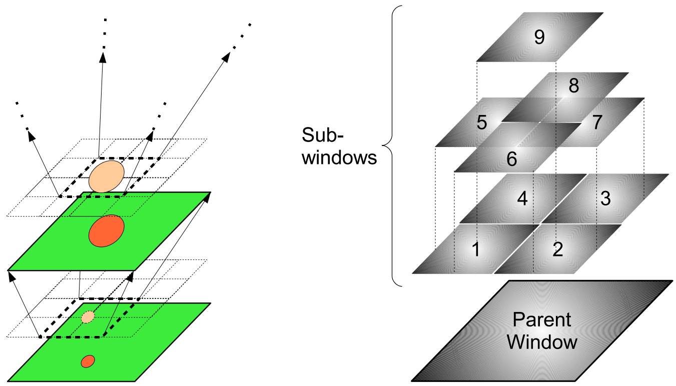
2.4 Overlapped Binary Search

So far we have defined a method to determine the areas in the image that do not cover an entire ball. However, this does not imply that an area contains a ball if it is not rejected using this measure. Furthermore, it does not guarantee that if the ball exists in the area, it is the only object surrounded by the area. In this section a recursive function is suggested to find the best ball candidates using the edge orientation histogram measure.

The function is presented in algorithm 1.. It examines the given window using the above described method. The function first verifies if the window could contain a ball by scanning the histogram. If the window is not rejected it is divided into several overlapping sub-windows, which are recursively processed by the algorithm 1.. The base cases of the recursion are windows, which either have more than a certain number of zero components in their histogram or are smaller than the ball is expected to be. This is demonstrated in figure 5a.

Thanks to the overlapped search algorithm, no problem occurs when the ball lies on the border of neighboring sub-windows. The object can be entirely covered by at least one sub-window. As demonstrated in figure 5b, the sub-windows are a quarter of the area of the parent window and are distributed both horizontally and vertically, each one fourth of the parent window edge length. A window is thus divided into 9 sub-windows.

The overlapped searching can reference a window more than once. We therefore suggests a look up table with an element for each possible window, down to the desired depth storing the search result for that window. A repeated reference can be detected at the beginning of the function and replied with the pre-stored search result.



(a) Recursive Binary Search (b) Overlapping Sub-windows

Fig. 5. Overlapped Binary Search: (a) The search recursion focuses on windows containing possible ball candidates and goes deeper until the size of the widow becomes smaller than the size of the ball. (b) A window is divided to 9 overlapping sub-windows.

2.5 Outlier Elimination

In accordance with algorithm 1., a ball candidate is found if all sub-windows of an accepted parent window are rejected to contain an entire ball. However the results are still subject to false positives. It is therefore required to further filter the output of the algorithm using a geometrical criterion. Two measures are suggested, both of which can be obtained from the histogram so that no further reference to the image is needed.

The first measure computes the standard deviation and the average of the histogram. The following condition verifies how uniform the distribution is.

$$\sigma < \alpha\mu \tag{5}$$

where σ is the standard deviation and μ is the average of the histogram. α is a constant, which determines the accepted uniformity of the distribution. The higher α becomes, the more candidates are accepted as balls. We set $\alpha = 2.0$ for our scenario.

The second measure verifies whether the number of edge points the window contains matches the circumference of the window. Assuming d to be the edge length of the window and n the number of edge pixels found in the window, the following is the condition to find balls.

$$\frac{\pi}{2}d(1-\beta) < n < 4d(1+\beta) \tag{6}$$

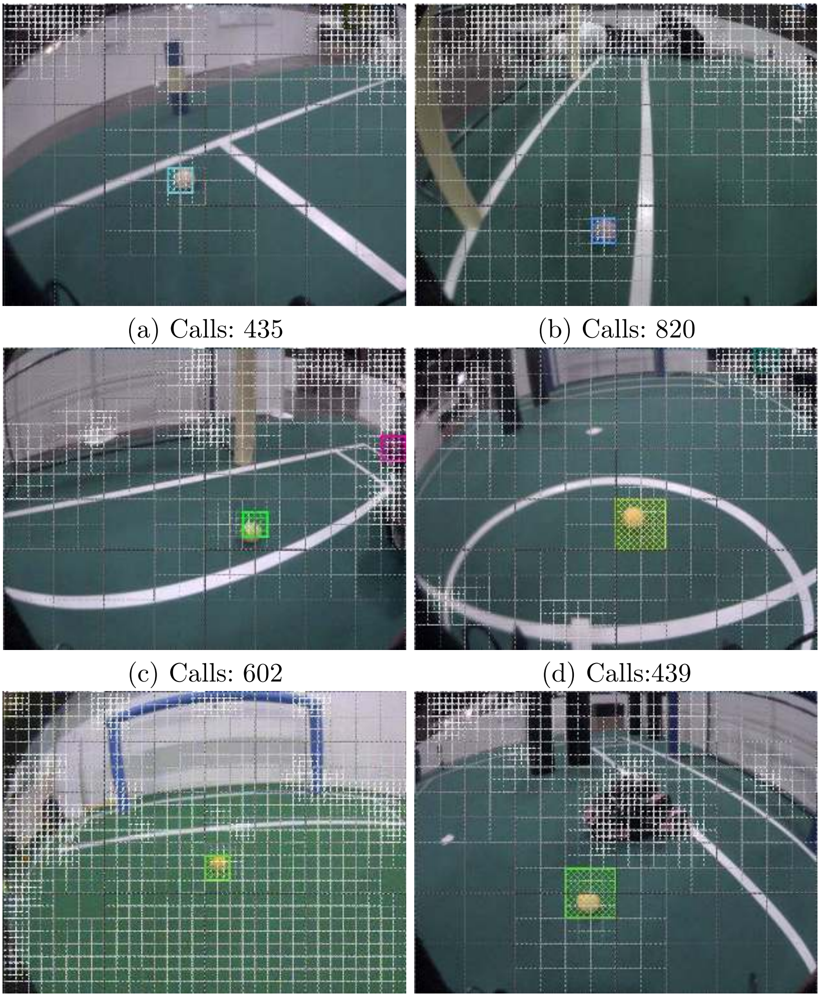
As window size is halved in each level, the window can be up to double the ball size. This sets the lower bound of the pixel count. The upper bound is set to the circumference of the window. β is an adjustable tolerance added to the condition. As a prerequisite to this condition the edge thickness should be reduced to one pixel using non-maxima-suppression.

2.6 Results

The algorithm was tested off-line using images recorded from RoboCup humanoid field. A data base of 3500 images from the recordings during RoboCup 2010 is used for the evaluation of the algorithm. The dataset includes images without balls, images including one ball as well as images containing more than one ball. Table 1 summarizes the evaluation results. As a performance measure, the number of references to the recursive function is counted. Comparing this to the number of pixels of the image (in this case 512x512) this shows a promising optimization of the search method. However, the initial image scan to calculate the *HII* should also be considered but authors intend to implement this stage into the hardware. Some example results are presented in figure 6.The results are overlaid with windows that the algorithm has accessed, which shows how the method approaches the image.

Table 1. Evaluation of the Ball Detection Algorithm

Property	Value	Unit
Number of Evaluation Images	3500	-
Total Number of Balls in Ground Truth (N)	2872	-
Detection Rate (TP/N)	88	%
False Positive Rate (FP/N)	9	%
False Negative Rate (FN/N)	12	%
Average recursive function calls	673	Calls/Frame



(a) Calls: 435 (b) Calls: 820

(c) Calls: 602 (d) Calls:439

(e) Calls: 1338 (f) Calls:658

Fig. 6. Results of Ball Detection using Histogram of Edge Orientations. Images are overlaid with the windows checked recursively by the algorithm. Detected ball candidates are presented as colored windows.

3 Conclusion

In this paper, we have presented a method for shape based round object detection using a edge orientation histogram. It can be used to globally detect the ball in the RoboCup soccer scenario, based only on the shape of the ball and is completely independent on factors like color and brightness. The image is represented using a histogram of the orientation of gradient vectors in each pixel. The representation can be implemented in hardware and accelerates the search algorithm considerably. A recursive search algorithm is then used to detect the ball in the image in relatively few function calls, compared to the number of pixels in the image. A statistical measure is used to eliminate most outliers. This method is fast due to few references to the image, and robust, as it has very few false positives.

References

[1] Bonarini, A., Furlan, A., Malago, L., Marzorati, D., Matteucci, M., Migliore, D., Restelli, M., Sorrenti, D.: Milan Robocup Team 2009. In: RoboCup 2009 Graz, Graz, Austria (2009)
[2] Canny, J.: A computational approach to edge detection. IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-8(6), 679–698 (1986)
[3] Coath, G., Musumeci, P.: Adaptive arc fitting for ball detection in robocup. In: Proceedings of APRS Workshop on Digital Image Analysing, Brisbane, Australia, pp. 63–68 (2003)
[4] Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: Proc. IEEE Computer Society Conf. Computer Vision and Pattern Recognition, CVPR 2005, vol. 1, pp. 886–893 (2005)
[5] Hanek, R., Beetz, M.: The contracting curve density algorithm: Fitting parametric curve models to images using local self-adapting separation criteria. International Journal of Computer Vision 59, 233–258 (2004)
[6] Li, X., Lu, H., Xiong, D., Zhang, H., Zheng, Z.: A Survey on Visual Perception for RoboCup MSL Soccer Robots. International Journal of Advanced Robotic Systems 1 (2013)
[7] Martins, D., Neves, A., Pinho, A.: Real-time generic ball recognition in RoboCup domain. In: Proc. of the Ibero- . . . , 11th edn. (2008)
[8] Mitri, S., Frintrop, S., Pervolz, K., Surmann, H., Nuchter, A.: Robust object detection at regions of interest with an application in ball recognition. In: Proceedings of the 2005 IEEE International Conference on Robotics and Automation, ICRA 2005, pp. 125–130 (April 2005)
[9] Neves, A.J., Pinho, A.J., Martins, D.A., Cunha, B.: An efficient omnidirectional vision system for soccer robots: From calibration to object detection. Mechatronics 21(2), 399–410 (2011)
[10] Roberts, L.G.: Machine perception of three-dimensional solids. Technical report, DTIC Document (1963)
[11] Treptow, A., Zell, A.: Real-time object tracking for soccer-robots without color information. Robotics and Autonomous Systems 48(1), 41–48 (2004)
[12] Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: Proc. IEEE Computer Society Conf. Computer Vision and Pattern Recognition, CVPR 2001, vol. 1 (2001)