Robust Real-Time Detection of Multiple Balls on a Mobile Robot

Andreas Masselli* Richard Hanten* Andreas Zell*
*Department of Computer Science, University of Tuebingen, Tuebingen, Germany

***Abstract*— In this paper, we present a robust method for detecting multiple balls without using color information. The method builds upon the boosted Haar classifier approach introduced by Viola et al. [11] and is applied onto an image from a catadioptric camera without performing rectification of the image. We compare the method with one based on a classical color segmentation approach. Originally both methods have been developed for the participation on the 2012 ”SICK robot day”, an international robotics competition. The task was to collect balls with a robot and put them into dedicated perches within a certain amount of time. Both methods showed to be robust during the competition. Our team placed second out of fifteen contenders.**

I. Introduction

Real-time detection of dynamic objects is still an interesting research topic in the field of robotics. Many robotics competitions involve object detection tasks, like the RoboCup, ELROB, US FIRST, or FIRA RoboSot[1]. Often objects like balls are labeled with a specific signal color, since it is challenging to implement algorithms that are independent of color information. While still popular, the goal is to move away from color labeling and use objects of arbitrary color, which enlarges the field of application of the used methods. In the ”SICK robot day”, an international robotics competition, white balls have been introduced. Methods for detecting ball objects without color information have been presented by Hanek et al. [5] or Treptow et al. [10] within the context of the RoboCup robotics competition. Both methods share the assumption, that there is only a single ball to detect. They use particle filters or other tracking approaches that follow a single object in the image and fail once more objects are introduced. In this paper we present a method based on the approach by Viola et al. [11] which is able to detect multiple balls in real-time. The method does not rely on color information and has proven to be robust against different lighting conditions. Since it works without any tracking, it can be used in a highly dynamic environment with non linear motion.

We successfully applied the method in one run of the ”SICK robot day”.

Although the detection via the Viola-Jones approach is not rotationally invariant by default, we were able to apply it on the raw image of a catadioptric camera without performing a cylindrical or spherical projection, like it is done e.g. in [3], [6] or [9]. This way we enabled real-time processing without the need of additional graphical hardware. To our knowledge this is the first time the classifier by Viola et al. is utilized in this manner.

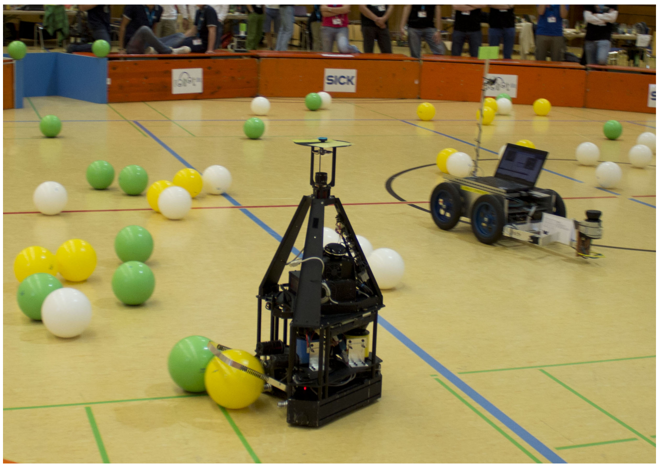

Fig. 1. Our robot picking up a ball during the robotics competition.

The remainder of this paper is organized as follows: In Section II we describe the 2012 SICK robot day robotics competition. Section III gives details of our experimental setup. The ball detection approaches that we implemented are explained in Section IV. In Section V the experimental results of both methods are presented and discussed. We conclude our paper in Section VI.

II. SICK Robot Day Competition

The ”SICK Robot Day” is an international robotics competition for research teams organized by the sensor company SICK AG. It takes place in Waldkirch, Germany every two years. Each time a new task is introduced. In 2012, the challenge was to collect rubber balls with a single robot and put them into a dedicated perch. 15 teams participated by designing and programming an autonomous mobile robot dedicated to fulfill the task. The indoor competition featured a circular arena with a diameter of approx. 20 m, which was filled with 87 balls of three different colors, 29 green, 29 yellow and 29 white ones. The balls had a diameter of 18 cm. Each team participated in two 10 minute runs: During one run, three teams competed simultaneously with their robot, each being assigned to a specific ball color drawn by lot. Starting at a home perch, which was marked with the team's ball color, the robot could earn one point by dropping a ball with the assigned color into the perch, or lose one point

[1] RoboCup: www.robocup.org, ELROB: www.elrob.org, US FIRST: www.usfirst.org, FIRA RoboSot: www.fira.net/?mid=robosot

instead for dropping a ball of a different color.

While it is common to use a color segmentation to detect balls in such competitions, like in the RoboCup, here a difficulty arose to detect the white balls. They do not contain color information which one could use to distinguish them from parts in the camera image that are less colored as well. Brightness alone is insufficient for detection, due to lighting changes and shadows of the balls. Sun glares and reflection of sunlight from the large windows at the arena also appear as white spots in the camera image. We therefore implemented a method for detection of white balls comprising shape information, and compared the method with a classical color segmentation approach.

III. Experimental Setup

A. Robot Setup

The experiments were accomplished using two custom built robots that were former used as soccer players in the RoboCup. They feature a triangular basis with holonomic drive using three *Swedish wheels*, a ball kicking device, and a gripper to pick up balls, especially designed for participating at the SICK robot day. The robots are equipped with two laser rangefinders, one SICK S300 and one LMS100. They allow the robots to scan almost all of the arena. An omnidirectional camera is mounted on top of the robot, looking upwards against a hyperbolic mirror. The camera itself is an AVT Marlin with a resolution of 780×580 pixels, working at a frame rate of 15 Hz. It is featured by a modified driver which allows us to directly work with *YUV-422* images (8 bits per channel). All of the application code for the task runs on an onboard Mini-ITX computer featuring an Intel Core 2 Duo with a processing speed of 2.53 GHz and 2 GB of RAM. We budgeted one core for image processing, the other core was needed for the remaining modules such as localization and path planning.

B. Datasets

During the time working on the challenge 37 log files were captured. They were recorded in several indoor places, so the algorithms properties could be trained and tested under changing environments. For training the Viola/Jones approach nine log files were used to create patches containing white balls. One log file was taken directly during the SICK robot day from within the original arena, containing 866 images. Data from this file was used as validation set in our experiments presented in section V. This way we could examine the performance of both approaches under real competition conditions, and demonstrate the robustness of the detectors when run in a new environment.

IV. Detection Methods

The following section describes the implemented methods to detect the balls within the camera image. A typical image is shown in Figure 2. This section is split in two parts: First we explain our implementation of the classical color segmentation approach. In the second part we describe how we applied the object detection framework by Viola et al. [11] to the problem.

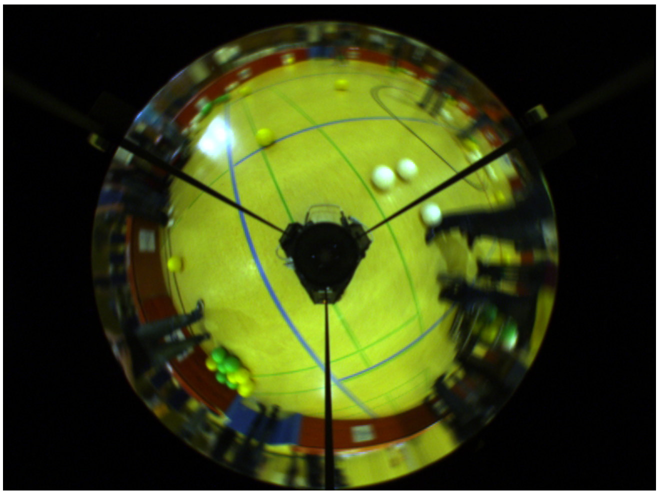

Fig. 2. Typical image from the catadioptric camera.

A. The Color Segmentation Approach

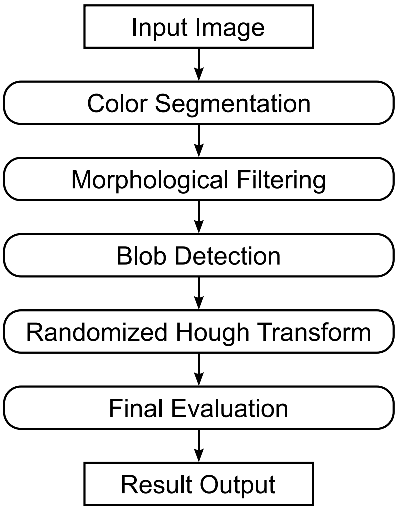

Fig. 3. Image processing pipeline of the color segmentation approach.

Our first approach relies on many readily-available methods, e.g. on the OpenCV graphics library [1]. Figure 3 displays the general pipeline which is used to process the images. For each color of interest, there exists a binary image to which every positively tested pixel position is written. After generating the binary images, they are filtered to remove speckles from noise while keeping regions of interest. These remaining regions are found using the *cvBlobLib* library [7]. The library determines the connected regions, or *blobs*, in each binary image and returns their contour. On these contours we apply a *Randomized Hough Transform* [12] to detect circular shapes. Every shape which can be approximated with a circle is considered a ball and put to the results. The remaining blobs are discarded.

1) Color Segmentation: Each color segmentation requires defined color ranges or discrete values within a color space. We chose YUV as color space, since the camera provided YUV images, saving time consuming image conversions.

Color values could either be ignored or picked. We sampled colors from the training set described in section III-B for each ball color and defined all colors within a radius of 15 within the color space as ball color. These colors are stored in a *lookup table* to speed up the query process during the segmentation.

2) Morphological Filtering: With assuming a ball size of at least four pixels within the image, we remove noise by applying morphological filters to the binary images. Omitting this step would slow down the entire detection, since too many regions in the image would be further examined by the following detection steps. The operators that are used on the images are the *open* and the *close* operator given by the OpenCV framework.

3) Blob Detection: In the next step connected regions are found on the filtered binary images. It is implemented using the cvBlobLib, which returns a list of polygons for each region. A typical segmented image can be seen in figure 4.

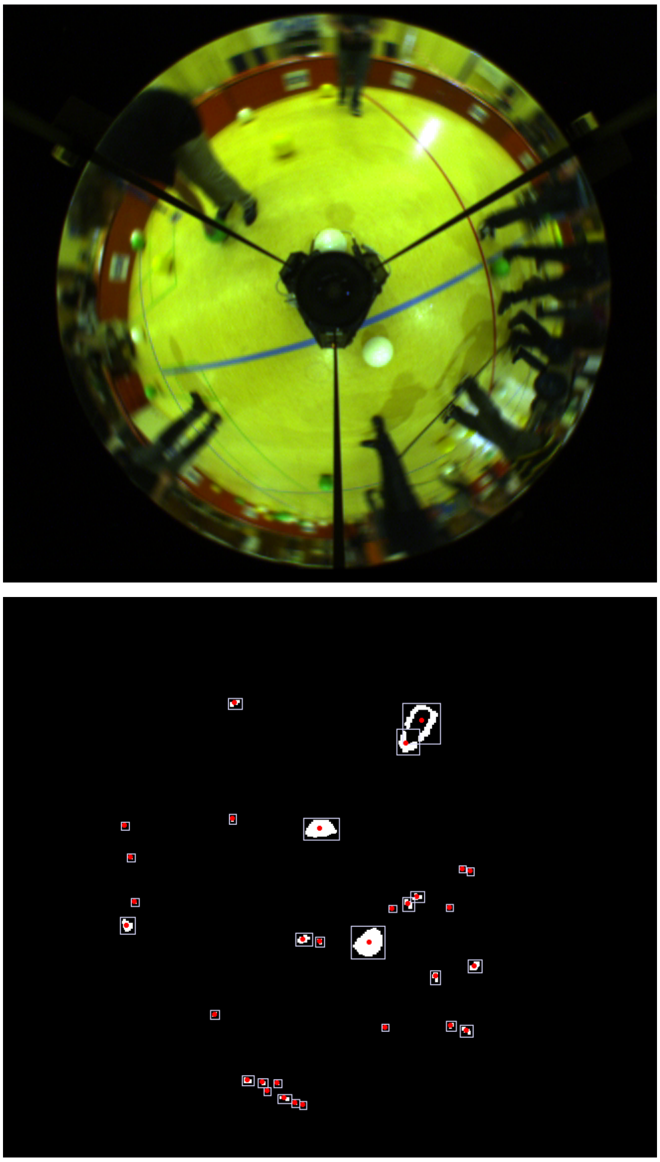

Fig. 4. Blob extraction using OpenCV.

4) Randomized Hough Transform: The extracted blobs are examined regarding the shape using the Randomized Hough Transform (RHT) [12]. The balls appear circular in the image, therefore blobs that are not recognized as circular by the RHT are rejected and not further processed. Since there are many balls, they often occlude each other, and several balls appear as one single blob. This is why we chose the RHT over other common methods like RANSAC [4], since it can detect multiple circles (see figure 5).

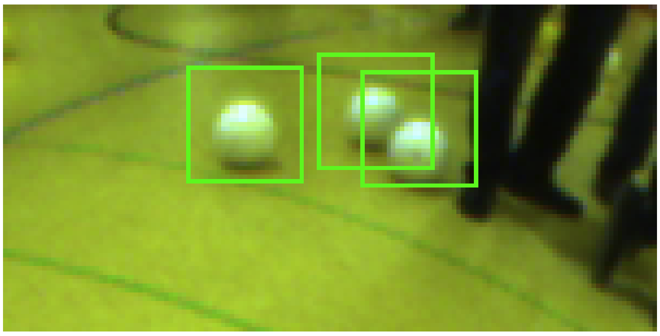

Fig. 5. Detection of occluding balls.

Every time circles are merged, the position and radius of the already existing one are corrected. Let $\vec{c}_{new}$ and r_{new} be the new calculated center and radius while $\vec{c}_{old}, r_{old}$ are already known and $\vec{c}_{found}, r_{found}$ are corresponding to a new estimated circle.

$$\vec{c}_{new} = \frac{v}{v+1}\vec{c}_{old} + \frac{1}{v+1}\vec{c}_{found}$$

$$r_{new} = \frac{v}{v+1}r_{old} + \frac{1}{v+1}r_{found}$$

Besides correcting the position, the circle is voted to be more important by counting the times it was merged with newer circles. Meaning that it is very likely that there actually is a circle at this position. Depending on the times triples are sampled, it is possible to define a minimum amount of votes, so a circle is accepted to be valid. If there is no valid circle no clear circular shape could be estimated.

5) Final Evaluation: We perform a final test on the remaining regions of interest to remove false positives. Pixels with the color of interest given by the *lookup table* are counted within the circle. After that the amount of these pixels was put in relation to the estimated area of the circle. Objects with a ratio of the desired color below a given threshold are rejected.

B. The Boosted Haar Classifier Cascade Approach

To overcome the problem of lacking color information and exploiting the shape features of a ball in the camera image, we implemented a detector based on the approach by Viola et al. [11]. It uses Haar wavelets as basic visual features, shown in figure 7. They capture information of intensity differences of neighboring regions in a grayscale image, which we directly get from the camera using the Y component of the YUV image. The Haar features are scalable and can be evaluated quickly, utilizing an *Integral Image*. Originally designed for face detection, it makes use of a training database consisting of face and non-face images. Viola et al. apply *AdaBoost* on the database, a supervised machine learning algorithm, which selects a set of Haar features to distinguish between faces and non-faces. The so trained classifier can then be used to detect faces in an image by applying it on a subwindow, which is moved across the image line by line and in multiple scales.

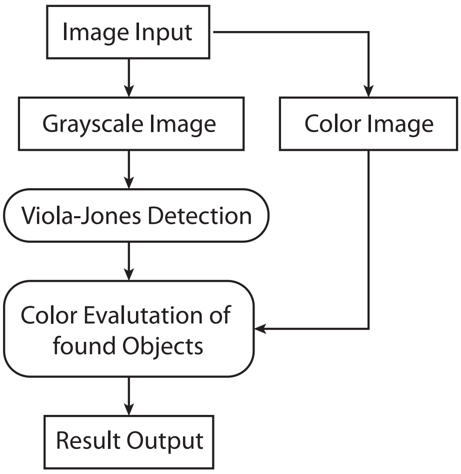

Fig. 6. Image processing pipeline of the Haar classifier approach.

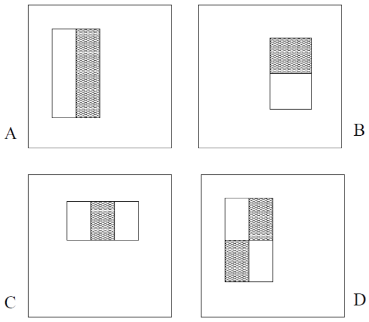

Fig. 7. The four used types of Haar features (from [11]).

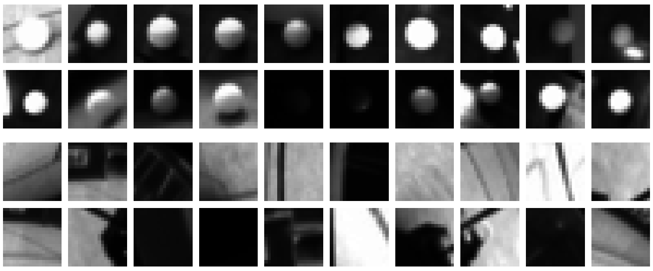

Fig. 8. Example image patches used for training the classifier.

To speed up the detection process, a so called *Attentional Cascade* is introduced [11]. A cascaded classifier is built from a series of classifiers, acting as a degenerated decision tree: Starting with the first classifier from the cascade, the succeeding classifier is only evaluated if the preceding classifier has not rejected the image (see figure 9). This way an image can quickly be scanned for candidate regions of a face with a fast but unspecific classifier, and subsequently examined on the remaining candidate regions by more and more sophisticated classifiers.

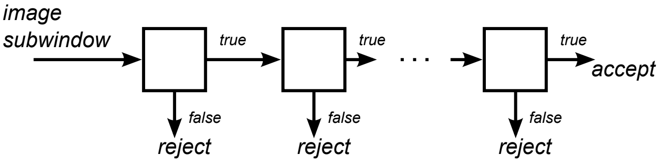

Fig. 9. Structure of the attentional cascade.

Depending on the training set, the Haar classifier method can be used to detect arbitrary objects, which has been shown to work with soccer balls [10] and in a former SICK robot day competition with numbered signs [8]. Due to their rectangular shape, Haar features may seem unsuitable for detecting circles. However, since they can be evaluated rapidly, many of them can be used to approximate round shapes.

Compared to RoboCup [10], we experienced the given scenario to be more challenging in two aspects: First, compared to a soccer ball, the white balls showed no texture at all (see fig. 2), second, there were multiple balls in the arena, whereas in [10] it was assumed that there is only one soccer ball on the field. This assumption allowed the use of a particle filter, which speeded up the detection and could compensate wrong classifications of the cascade.

We created a training set based on log files recorded with our robot (see also section III-B). For the non-ball images we simply cropped random patches out of an image series, which we recorded during a robot run through an arena without balls. The ball images were semi-automatically extracted using the differential image method: While the robot was not moving, single balls were rolled through the arena. These could be distinguished from the non moving background. After that the patches were reviewed, and false detections were removed. The final training set contained 2569 ball images and 10000 non-ball images. Example images are shown in figure 8. During training the cascade layers, we applied the bootstrapping technique from [11], i.e. we refined the training set after training one layer in two steps: First we removed all images that are rejected by the cascade that has been trained so far. In the second step, we applied the currently trained cascade on the image series from the run through the empty arena, and refilled the non-ball set with patches that pass the cascade, therefore being false positives which still need to be rejected by the succeeding layers. This way we maintained a large non-ball training set of 10000 images. For each cascade layer we set the detection rate to 0.97 and the false positive rate to 0.4. We trained until the cascade did not find enough false positives to refill the training set, which yielded a cascade consisting of 14 classifier layers, the first layer using 3 and the last using 300 features.

1) Reducing the Search Space: Despite the speed improvement from the attentional cascade, the approach by Viola et al. is slow compared to color segmentation. To achieve real-time performance, we made several adaptions, which resulted in an efficient detector. First we applied a fixed mask to the image from the catadioptric camera, i.e. we examined only the parts of the image that actually contained visual information. Second we considered the possible locations of balls in the image. They fully cover the masked region, and balls also appear in different sizes, which usually leads to the need of scanning the image several times with differently scaled Haar classifiers. However, since all balls lie on the ground during the competition, the size of a ball in the image functionally depends on its position within the image. The function is found by calibration, and helps to reduce the search space by one dimension. With these improvements we could reduce the number of image subwindows that need to be examined from originally 822,417 down to 97,576.

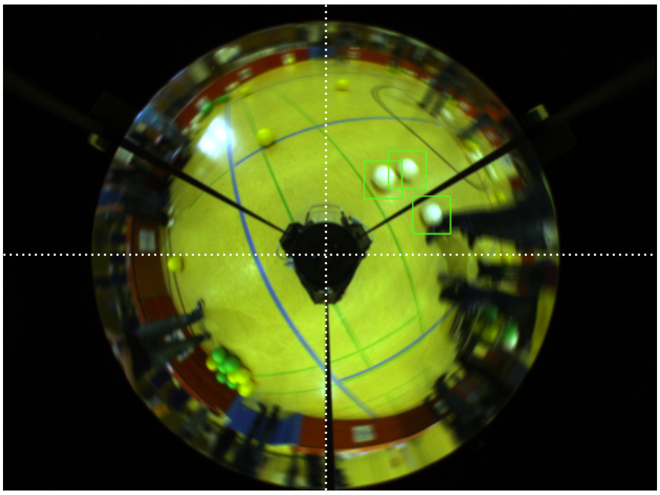

Fig. 10. The image from the catadioptric camera is subdivided into four quadrants, each scanned with a classifier specific for the respective ball orientations.

2) Handling Different Orientations: The Haar classifier is scalable, but not invariant to rotation, neither in plane nor out of plane. Since balls are round, one could think this is not an issue, but due to shading and shadows of the ball its image has an orientation. The top is always brighter than the bottom. This makes it necessary to regard orientation when designing the detector. In our case, images are taken from a catadioptric camera (see figure 2). Therefore the balls appear in every possible orientation. Many works exist, where the image is unwrapped using a spherical or cylindrical transformation, such that objects appear upright and allow detection using a Haar based classifier ([3], [6], [9]). In order to save computational time, we omit unwrapping the image, and find the variously oriented balls directly on the raw image.

To detect all orientations with a Haar classifier approach, one can train many classifiers for different orientations and apply them all onto the image. This was done e.g. in [2]. The drawback is that now the image is scanned multiple times with n detectors, therefore being n times slower than with a single one.

Another approach is to train one classifier with all possible orientations in the training set. This is possible if the objects remain similar in different orientations, as it is in our case. We therefore trained a classifier with balls of all orientations. This led to a working but slow classifier, because it learned a huge set of features (3954 features in the first ten cascade layers), which were needed to circumscribe the positive set within the feature space.

However, there is also a functional dependence of orientation and position in the image. Since the balls cast their shadows to the ground, these shadows always appear directed towards the center of the image (see figure 2). In the same manner the brighter part is directed away from the center. Therefore, the idea is to train a set of rotational specific detectors, and apply on each area only the detector which is specific for that orientation. We compromised this idea to train a classifier specific to all orientations that appear in the upper left quadrant of the image, and created the detectors for the remaining 3 quadrants by mirroring the features of the trained one. During the extraction of balls for the training we also mirrored the ball patches, that were not cropped out of the upper left quadrant of the camera image, such that all balls appear being extracted from the upper left quadrant.

With this technique we came up with a set of faster classifiers, that together cover the whole area, being split in four quadrants (see figure 10). While the first ten layers of the rotational invariant detector consisted of 3954 features, we reduced this number to 1116 for each rotational specific detector, without having the overhead of applying multiple detectors one each subwindow of the image.

3) Constant Time Approach: Because of the nature of the attentional cascade, the time for evaluating a camera image is not constant. It increases with the number of balls as well as with objects that appear similar to balls, since all these objects will pass the lower layers of the cascade and will trigger the evaluation of the more complex layers. This could lead to time lags where camera frames will be skipped, which is critical during the competition. To ensure that the algorithm runs in real-time, we implemented a timeout such that the algorithm stops scanning after a fixed time has passed, and started the scan close to the center of the image, effectively searching for balls close to the robot first. If the algorithm is stopped, it is likely to return a list of balls which are nearby. This was sufficient for the task during the ”SICK robot day” competition, since within the rules it is a good strategy to pick up balls close to the robot, especially if it is holonomic like in our case.

4) Color Evaluation of Found Objects: After detection of the balls, we added a final stage to the whole classifier which labels the balls according to their different color, since it is crucial for the task to pick up the balls with the right color. After passing the final layer of the cascade detector, the remaining patches are examined using the full YUV image from the camera and the *lookup table* described in the first approach. This way we could separate white balls

TABLE I

Detection performance of the Color Segmentation and Viola-Jones approach. The first two values are relative to the total number of true balls in the test set.

	Color Segmentation	Viola-Jones
true positives in %	51.1	63.0
false negatives in %	48.9	37.0
false positives per frame	1.73	0.45

from yellow balls, since they appear similar in the grayscale image.

V. Experimental Results

We evaluated the performance of both approaches as well as their computation time by using the ground truth dataset described in section III-B.

A. Detection Performance

The performance of the detectors was evaluated in the following way: Each ball is either correctly detected (true positive) or missed (false negative). Detections that do not coincide to a ball are counted as false positives. Hereby we say that a detection coincides with a ball if and only if the intersection of both bounding boxes covers at least one fourth of each individual bounding box. This is because the bounding boxes returned by the detectors usually differ from the manually drawn boxes slightly in position and size. The results are listed in table I.

While the detection rate of both methods may seem low to the reader, one has to consider that due to the catadioptric system the balls appear small in the image, even at distances of 1 m. Many images also contained motion blur. Therefore distant balls in blurred images were almost unrecognizable even for humans. This fact was tolerable for the competition, since it was important to find balls nearby.

B. Computation Time

We measured the computational time of both methods by running them on a single core of our robot (Core 2 Duo, 2.53 GHz, 2 GB RAM) without any other job running at the same time. For evaluating the Viola/Jones approach we did not use the constant time approach.

The color segmentation approach took 16.5 ms on average per frame, with a standard deviation of 4.2 ms, while the Haar based approach took 39.3 ms, with a standard deviation of 14.8 ms.

VI. Conclusions

We implemented and compared two methods for detecting multiple balls in a dynamic scenario. The classical approach showed to be fast and worked sufficiently for balls with a distinguishable color. For white balls our method based on a Haar classifier outperforms the classical approach. With this method our robot was able to detect white balls in real-time. Still it leaves enough computational resources for other processes that run on the mobile robot. This method is also transferable to other objects, since it is based on a machine learning approach that can be trained with arbitrary objects. While the Viola-Jones framework has already been applied on omnidirectional image data, we showed that it can work directly on unwrapped images. Assuming objects to be upright in a rectified image, it implies that these objects appear in a radial orientation within the unwrapped image, and therefore our technique can be applied.

For future work it would be interesting to further pursue the handling of in plane rotation, as it usually appears in images taken from omnidirectional cameras. One could arrange a set of eight detectors specific for each octant easily by rotating all features of a classifier in addition to mirroring. An even finer grading could be accomplished by performing several training rounds, each time with a training set of balls being rotated on plane towards a given orientation. While consuming more training time and disk space for all detectors, it will not increase processing time as still only one detector is applied onto a certain region in the image.

Acknowledgement

The authors would like to thank all other team members who participated in the ”SICK robot day” competition.

References

[1] G. Bradski. The OpenCV Library. *Dr. Dobb’s Journal of Software Tools*, 2000.

[2] Shaoyi Du, Nanning Zheng, Qubo You, Yang Wu, Maojun Yuan, and Jingjun Wu. Rotated haar-like features for face detection with in-plane rotation. In Hongbin Zha, Zhigeng Pan, Hal Thwaites, Alonzo C. Addison, and Maurizio Forte, editors, *VSMM*, volume 4270 of *Lecture Notes in Computer Science*, pages 128–137. Springer, 2006.

[3] Yohan Dupuis, Xavier Savatier, Jean-Yves Ertaud, and Ghaleb Hoblos. A framework for face detection on central catadioptric systems. In *ROSE*, pages 57–62, 2010.

[4] Martin A. Fischler and Robert C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, June 1981.

[5] Robert Hanek, Thorsten Schmitt, Sebastian Buck, and Michael Beetz. Towards robocup without color labeling. In Gal A. Kaminka, editor, *RoboCup 2002: Robot Soccer World Cup VI*, pages 179–194. Springer Berlin Heidelberg, 2003.

[6] Redouane Khemmar, Jean Yves Ertaud, and Xavier Savatier. Article: Face detection and recognition based on fusion of omnidirectional and ptz vision sensors and heteregenous database. *International Journal of Computer Applications*, 61(21):35–44, January 2013. Published by Foundation of Computer Science, New York, USA.

[7] Cristóbal Carnero Li nán. cvBlob. http://cvblob.googlecode.com.

[8] Sebastian A. Scherer, Daniel Dube, Philippe Komma, Andreas Masselli, and Andreas Zell. Robust real-time number sign detection on a mobile outdoor robot. In *Proceedings of the 6th European Conference on Mobile Robots (ECMR 2011)*, Örebro, Sweden, September 2011.

[9] Wolfgang Schulz, Markus Enzweiler, and Tobias Ehlgen. Pedestrian recognition from a moving catadioptric camera. In Fred A. Hamprecht, Christoph Schnörr, and Bernd Jähne, editors, *DAGM-Symposium*, volume 4713 of *Lecture Notes in Computer Science*, pages 456–465. Springer, 2007.

[10] André Treptow, Andreas Masselli, and Andreas Zell. Real-time object tracking for soccer-robots without color information. In *European Conference on Mobile Robotics (ECMR 2003)*, pages 33–38, Radziejowice, Poland, 2003.

[11] Paul Viola and Michael Jones. Robust real-time object detection. In *International Journal of Computer Vision*, 2001.

[12] L. Xu, E. Oja, and P. Kultanen. A new curve detection method: randomized hough transform (rht). *Pattern Recogn. Lett.*, 11(5):331–338, May 1990.