20th RoboCup International Symposium, Leipzig, Germany, July 2016.

Real-Time Visual Tracking and Identification for a Team of Homogeneous Humanoid Robots

Hafez Farazi and Sven Behnke

Autonomous Intelligent Systems, Computer Science Institute VI
University of Bonn, Germany
{farazi,behnke}@ais.uni-bonn.de
http://ais.uni-bonn.de

Abstract. The use of a team of humanoid robots to collaborate in completing a task is an increasingly important field of research. One of the challenges in achieving collaboration, is mutual identification and tracking of the robots. This work presents a real-time vision-based approach to the detection and tracking of robots of known appearance, based on the images captured by a stationary robot. A Histogram of Oriented Gradients descriptor is used to detect the robots and the robot headings are estimated by a multiclass classifier. The tracked robots report their own heading estimate from magnetometer readings. For tracking, a cost function based on position and heading is applied to each of the tracklets, and a globally optimal labeling of the detected robots is found using the Hungarian algorithm. The complete identification and tracking system was tested using two igus® Humanoid Open Platform robots on a soccer field. We expect that a similar system can be used with other humanoid robots, such as Nao and DARwIn-OP.

1 Introduction

Multi-target tracking is a well-known problem in computer vision, and has many applications, including traffic monitoring and automated surveillance. The aim of multi-target tracking is to automatically find objects of interest, assign a unique identification number to each, and to follow their movements over time. Multi-target tracking is fundamentally different to single-target tracking because of the difference in the state space model used for each. In particular, data association in situations of multiple detections with closely spaced and/or occluded objects makes multi-target tracking significantly more difficult. The expected number of visible targets is often unknown and may vary over time.

This work addresses a problem with an additional level of difficulty—the identification and tracking of multiple robots of identical appearance. Despite the lack of visual clues, our system is not only able to track each detected robot, but also to identify which robots are being tracked. This is done by generating a cost function for each tracklet, based on a motion model and the differences between the set of estimated and broadcasted headings of the robots. The output of the system, which is an estimation of the location and heading of each robot,

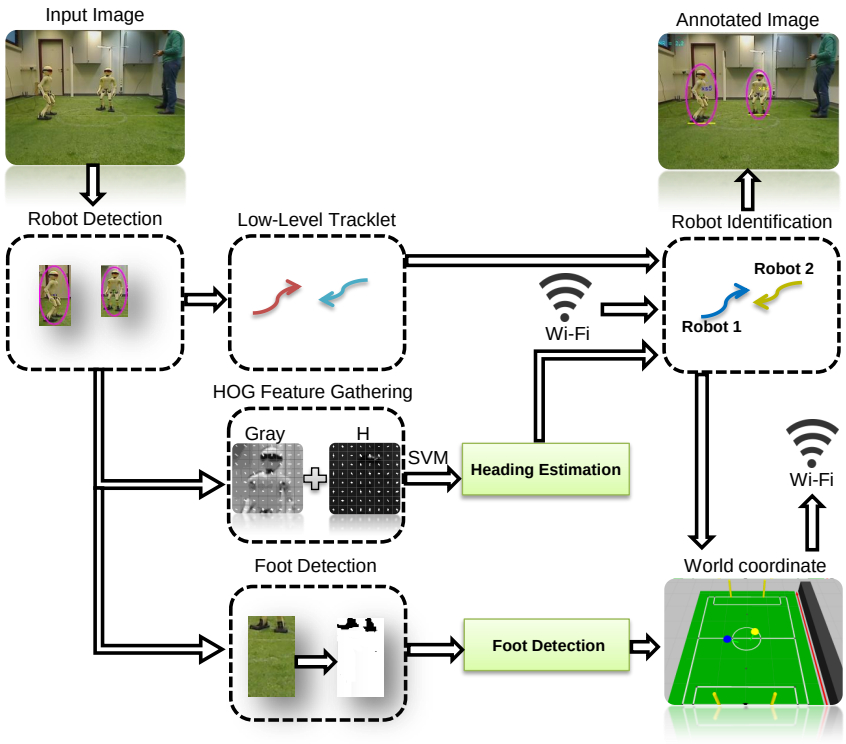

Fig. 1. Overview of our approach. After detection part, the heading of each robot is estimated based on proper HOG features. Using heading estimation and low-level tracklets observer finds and broadcasts the position of each robot.

is made available to the robots being observed, so that they may incorporate this into their own localization estimates, or use it for the generation of cooperative behaviors. Fig. 1 gives an overview of our system.
The main contributions of this paper include:

1. The introduction of a novel pipeline to identify a set of homogeneous humanoid robots in an image.
2. The development of a high accuracy and low training time humanoid robot detection algorithm, based on a Histogram of Oriented Gradients descriptor.
3. A robust method for the estimation of the relative heading of a robot.
4. Experimental evidence that the proposed method can cope with long-term occlusions, despite a lack of visual differences between the tracked targets.
5. Demonstration that it is possible to track, identify and localize a homogeneous team of humanoid robots in real-time from another humanoid robot.

2 Related Work

The related work is divided into three categories, *multi-target tracking*, *robot detection and tracking* and *visual orientation estimation*.

Multi-target tracking has been studied for many years in the field of computer vision. The tracking of targets in the absence of any category information is referred to as category free tracking (CFT) [23]. CFT approaches normally do not require a detector that is trained offline, but rely on manual initialization. Objects are tracked mainly based on visual appearance, and the system attempts to track each target by discriminating it from other regions of the image. The visual target model is usually updated online to cope with viewpoint and illumination changes. Two successful examples of the CFT approach include the works of Allen et al. [1] and Yang et al. [21]. Although CFT approaches are computationally inexpensive and easy to implement, they are prone to excessive drift, after which it is very hard to recover.

Tracking by detection is one of the most popular approaches to multi-target tracking problems, as objects are naturally reinitialized when they are lost, and extreme model drifts cannot occur. As such, association based tracking (ABT) methods, which associate object detections with observed tracks, are proposed e.g. by Xing et al. [20]. An offline training procedure is generally used for the detection of objects of interest in each frame, and continuous object detections over time are linked to form so-called tracklets. Tracklets can then be associated with each other to form longer tracks. In most works, the probability of two tracklets being associated with each other is calculated based on a motion model and other criteria of visual similarity. The global tracklet association of highest probability is then computed using either the Hungarian algorithm [20], a Markov chain Monte Carlo method [22], or a Conditional Random Field [14].

Robot detection and tracking was done by Marchant et al. [13] using both visual perception and sonar data which was targeted for soccer environment. However, anthropomorphic design requirements in the Humanoid League prohibit teams from using sonar sensors. Many object detection approaches cannot be used for robot detection tasks due to the limitation in the onboard computer. Arenas et al. [3] detected Aibo robot and humanoid robots using the cascade of boosted classifiers, which is suitable for real-time applications. In another work Ruiz-Del-Solar et al. [15] proposed nested cascades of boosted classifiers for detecting legged robots. In addition to robot detection, gaze-direction of the robot is estimated based on Scale Invariant Feature Transform (SIFT) descriptor by Ruiz-Del-Solar et al. [16].

Visual orientation estimation of an object is often done by comparing projections of an accurate 3D model of the object to what is observed in the image, and finding the orientation that best matches the detected features [7]. These approaches work best only on simple backgrounds. Since the background in our application can be quite cluttered, and we do not wish to rely on the existence of an accurate 3D model of the detected robot, the most suitable approach for orientation estimation is through the use of image descriptors. Lin et al. [11] proposed an orientation recognition system based on a SIFT descriptor [12], and a Support Vector Machine (SVM) classifier [5]. Shaikh et al. [18] proposed a template-based orientation estimation method for images of cars, based on the comparison of shape signatures.

3 Multi-Target Tracking Formulation

We assume to have a collection of N humanoid robots of identical appearance that need to be identified and tracked by a further standing robot, or a stationary camera. In each camera frame, each of the robots can either be fully visible, partially visible, or not visible at all, and may be performing soccer actions such as walking, kicking and getting up. As such, the durations of partial or total lack of visibility may either be short or long. Each robot is equipped with a 9-axis inertial measurement unit (IMU), and the estimated absolute heading of the robot is broadcasted over Wi-Fi. The Wi-Fi communication between the robots is assumed to have delays, data loss, and even potentially connection loss for up to a few seconds. We use NimbRo network library [17] for Wi-Fi communication. Our objective is to detect, track and identify the N robots based solely on the captured images and the broadcasted heading information. Two igus® Humanoid Open Platform robots were used for the verification of the approach in this paper.

4 Vision System

4.1 Robot Detection

Although a number of pre-trained person detectors are available online, there is no detector for humanoid robots that can work out of the box. As such, we have designed, implemented and tested a robot detector that can robustly detect the igus® Humanoid Open Platform, although we expect the detector to work for other humanoid robot model as well, with the appropriate retuning and retraining. We evaluated five different methods for their suitability in our target domain before selecting and refining the most promising one. The methods were based, respectively, on color segmentation [8], adaptive object labeling [19], Haar wavelets [10], Local Binary Patterns [9], and Histograms of Oriented Gradients (HOG) [24]. The last of the five, a HOG feature descriptor used in the form of a cascade classifier was chosen based on criteria such as detection rate and training time. Although many RoboCup teams use a simple color segmentation approach to robot and obstacle detection, this approach is not safe in our case because we want to be able to distinguish the igus® Humanoid Open Platform from other objects on the field, such as the referee. Adaptive object labeling produced a relatively high rate of false positives, and it was nearly impossible to find a suitable threshold to work at all distances and in all situations. The method was also not able to deal well with occlusions. The Local Binary Pattern-based feature classifier also produced relatively poor results. However, the overall detection rates for the Haar wavelet and HOG cascading methods were found to be relatively good, and quite similar, but the former required a significantly longer time to train, so the latter was chosen.

In contrast to what is suggested for pedestrian detection [6], we do not feed the output of a multi-scale sliding window to a support vector machine (SVM) classifier. Instead, to save on computation time we use a cascade of rejectors with the AdaBoost technique to choose which features to evaluate in each stage,

Fig. 2. Robot detection results under various conditions.

similar to what is suggested by Zhu et al. [24]. By using HOG features, we obtain a description of the visual appearance of the robot that is invariant to changes in illumination, position, orientation and background. As HOG is not rotation-scale invariant however, we artificially expand the number of positive images used for training by applying a number of transformations, also in part to minimize the required user effort in gathering the samples. These transformations include random rotations up to $\pm 15°$, mirroring, and the cutting of some parts of the sample image, in particular at the bottom, left and right, to emulate partial occlusion. Note that larger rotations of the images are not applied to allow the classifier to learn the shadow under the robot. This also has the positive effect of not detecting sitting or fallen robots, so that this discrete difference can be used in the identification phase to discern the robots. In our training of the igus® Humanoid Open Platform, we used a set of about 500 positive samples, 1000 negative samples, and a cascade classifier with 20 stages. The training time for the classifier was about 12 h on a standard PC.

As demonstrated in Fig. 2, this approach can detect the robot under various conditions, including while walking and kicking. The best detection results on the RoboCup field are at distances between 1 m and 5 m when the observer is not moving. After some post-processing, mainly related to non-maximum suppression, a bounding box for each detection is computed.

4.2 Heading Estimation

Given that all robots being detected in our application have the same visual appearance, estimation of the robot heading relative to the observer forms a primary cue to identify the robots, especially after long occlusions. To visually estimate the robot heading, we analyze the bounding boxes reported by the robot detector. We formulate the heading estimation problem as a multiclass classification problem by partitioning the full heading range into ten classes of size 36°, and use an SVM multiclass classifier with an RBF kernel.

The estimation is performed based on the output of a dense HOG descriptor on the upper half of the bounding box, the center position of the bounding box, and potential color features. The dense HOG features are used in the heading estimation to represent the visual features of each rotation class in the grayscale channel. Visual features of the robot are different depending on the position of the robot in the image. To address this issue, we pass the normalized position of the detected robot to our classifier. Many robots, including ours, have color features that can be used to help classify the robot heading. Hence, dense HOG features are also computed on the H channel, and the resulting feature vector is

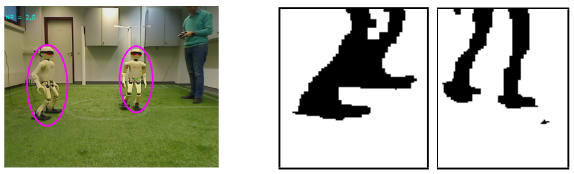

Fig. 3. Two non-green binary images for the detected robots in the left image.

forwarded to the SVM classifier. To acquire the best possible results from the classifier, implemented using the LIBSVM library [4], all feature data is linearly scaled to the unit interval, and k-fold cross-validation and grid searching was used to find the best parameter set.

4.3 Foot Detection

Once a robot has been detected, it is desirable to be able to project the position of the robot to the egocentric world coordinates of the observing robot. For this to be reliable, a good estimate of the lowest part of the detected robot is required. Due to the non-maximum suppression in use, the bounding box often may not include all pixels of the robot feet. Due to the high sensitivity of the projection operation, especially when the robot is far from the observer, this causes significant errors in the estimated robot distance. To overcome this problem, we make the assumption that the robot is located on a surface of a mostly uniform known color. Starting from an appropriate region of interest, and using erosion, dilation and color segmentation techniques, we construct a segmented binary image such as the one in Fig. 3. A horizontal scan line scheme is then applied to improve the estimate of the bottom pixel of the robot. In more complicated cases, outside of the context of RoboCup, in which it is not possible to rely on a single predefined field color, one could use a background-foreground classification approach similar to the one proposed in [14]. After building a probability image, where each pixel contains the probability that it belongs to the background, our proposed method can be applied.

5 Tracking and Identification System

Many previous works in the area of tracking and identification are not suitable for our application, because they either work offline or are too computationally expensive. In this work, we propose a real-time two-step tracking system that first constructs low-level tracklets through data association, and then merges them into tracks that are labeled with a robot ID based on tracklet angle differences and the reported robot heading information. For the low-level tracking, we use greedy initialization, albeit with the assumption that the new tracklet should not be in the vicinity of another existing tracklet, in which case lazy initialization ensures that the detection is a robot and not a false positive. We use lazy deletion to cope with occlusion and false negatives.

5.1 Kalman Filter

Kalman filters are a state estimation technique for linear systems, with the general assumption that process and observation noise are Gaussian. Many researchers utilize Kalman filters as part of their object tracking pipeline, mainly due to its simplicity and robustness. Kalman filtering involves two main steps: Prediction and correction. In each cycle, a new location of the target is predicted using the process model of the filter, and in every frame where we detect a target, we update the corresponding Kalman filter with the position of the detection to correct the prediction. Using this approach, the target can still be tracked even if it is not detected or occluded. We use the constant acceleration model to derive the predictions in our model, in the one-dimensional case:

$$\begin{aligned} p_{k+1} &= p_k + \dot{p}_k \Delta T + \tfrac{1}{2}\ddot{p}_k \Delta T^2, \\ \dot{p}_{k+1} &= \dot{p}_k + \ddot{p}_k \Delta T, \\ \ddot{p}_{k+1} &= \ddot{p}_k, \end{aligned} \tag{1}$$

where p_k, $\dot{p}_k$ and $\ddot{p}_k$ are the position, velocity and acceleration respectively at time step k. So, in our two-dimensional case the state vector becomes

$$\boldsymbol{x}_k = \begin{bmatrix} h_k & v_k & \dot{h}_k & \dot{v}_k & \ddot{h}_k & \ddot{v}_k \end{bmatrix}^T, \tag{2}$$

where (h_k, v_k) is the position of the center of the robot in the image at time step k. The system model is then given by

$$\boldsymbol{x}_{k+1} = \boldsymbol{\Phi}\boldsymbol{x}_k + \boldsymbol{w}_k, \tag{3}$$

where $\boldsymbol{w}_k \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{Q}_k)$ is zero mean Gaussian process noise with covariance $\boldsymbol{Q}_k$ and $\boldsymbol{\Phi}$ is the state transition matrix, derived from (1):

$$\boldsymbol{\Phi} = \begin{bmatrix} 1 & 0 & \Delta T & 0 & \frac{1}{2}\Delta T^2 & 0 \\ 0 & 1 & 0 & \Delta T & 0 & \frac{1}{2}\Delta T^2 \\ 0 & 0 & 1 & 0 & \Delta T & 0 \\ 0 & 0 & 0 & 1 & 0 & \Delta T \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \tag{4}$$

ΔT is the nominal time difference between two successive frames. In every frame where the robot is detected, the Kalman filter is updated using the coordinates of the center of the detected robot bounding box $\boldsymbol{z}_k = (\hat{h}_k, \hat{v}_k)$. The measurement model is given by

$$\boldsymbol{z}_k = \boldsymbol{H}\boldsymbol{x}_k + \boldsymbol{\vartheta}_k, \tag{5}$$

where $\boldsymbol{\vartheta}_k \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{R}_k)$ is zero mean Gaussian measurement noise with covariance $\boldsymbol{R}_k$ and $\boldsymbol{H}$ is the measurement matrix

$$\boldsymbol{H} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{6}$$

Given this system model, measurement model, and some initial conditions, the Kalman filter can estimate the state vector $\boldsymbol{x}_k$ at each time step together with its covariance $\boldsymbol{\Sigma}_k$.

5.2 Data Association

In multi-target tracking, the problem of finding the optimal assignment between new target detections and existing tracklets, in such a way that each detection is assigned to at most one tracklet, is referred to as the data association problem. Assume that in the current frame we have n existing tracklets, and m new detections, where m is not necessarily equal to n. Let $\boldsymbol{p}_i$ denote the predicted position of the i^{th} tracklet, and $\boldsymbol{d}_j$ denote the position of the j^{th} detection. We construct the $n \times m$ cost matrix $\boldsymbol{C}$, with entries given by

$$C_{ij} = \begin{cases} \|\boldsymbol{p}_i - \boldsymbol{d}_j\| & \text{if } \|\boldsymbol{p}_i - \boldsymbol{d}_j\| < D_{max}, \\ C_{max} & \text{otherwise,} \end{cases} \tag{7}$$

where $i = 1, \ldots, n$ and $j = 1, \ldots, m$, D_{max} is a distance threshold, and C_{max} is the length of the diagonal of the image in units of pixels. Using the cost matrix $\boldsymbol{C}$, the optimal data association is calculated using the Hungarian algorithm.

5.3 Robot Identification

We modeled the problem of identifying the robots as a high-level data association problem. In each time step, we have n tracklets and r robots, where r is determined by the observer as the number of robots that are broadcasting their heading information over Wi-Fi. Each tracklet, in addition to a buffer T_{pos} of (x, y) pixel position values, incorporates a buffer of detected robot headings T_{rot}. Buffers R_{rot} of received absolute headings from the robots are also maintained. The previously calculated robot positions are also kept in a buffer R_{pos} of pixel position values. We wish to optimally assign each tracklet to at most one robot, based on the detected and received heading information Fig. 4. The core idea is to find the best tracklet assignments based on the average of the differences between the detected tracklet heading buffers and the broadcasted headings from the individual robots over a limited time range. We construct the $n \times r$ cost matrix $\boldsymbol{G}$, with entries G_{ij} that relate to the cost of associating the i^{th} tracklet with the j^{th} robot:

$$\gamma = \begin{cases} \frac{r}{2\pi} \min\left\{ \left| R^a_{rot}[1] - R^b_{rot}[1] \right| : a < b,\ a, b \in 1, \ldots, r \right\} & \text{if } r \geq 2, \\ 0.5 & \text{otherwise,} \end{cases} \tag{8}$$

$$G_{ij} = \begin{cases} \frac{\gamma}{\pi D_i} \sum_{k=1}^{D_i} \left| T^i_{rot}[k] - R^j_{rot}[k] \right| + \frac{1-\gamma}{C_{max}} \left\| T^i_{pos}[1] - R^j_{pos}[1] \right\| & \text{if } D_i \geq \tau, \\ 2.0 & \text{otherwise,} \end{cases} \tag{9}$$

where τ is a minimum buffer size threshold, D_i is the number of elements in the buffers of the i^{th} tracklet, and for example $T^i_{rot}[k]$ is the k^{th} element of the T_{rot} buffer for the i^{th} tracklet, where $k = 1$ corresponds to the most recently added value, and $k = D_i$ corresponds to the oldest value still in the buffer. Similarly, $R^j_{pos}[1]$ is the most recent (x, y) coordinate in the R_{pos} buffer for the j^{th}

robot. The interpolation factor γ determines, based on the minimum separation of the broadcasted robot headings, how much we should rely on differences in heading to associate the robots, and how much we should rely on differences in detected position. Once the cost matrix $\boldsymbol{G}$ has been constructed as described, the Hungarian algorithm is used to find the optimal robot-to-tracklet association. With that association, all information that is required to compute the egocentric world coordinates of the detected robots relative to the observer is available. Some low-pass filtering is performed on the final world coordinates to reduce the effects of noise, and produce more stable outputs.

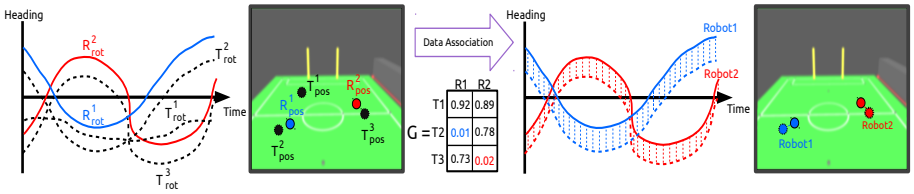

Fig. 4. Robot identification overview. We associate low-level tracklets with robots using comparison of heading and position.

6 Experimental Results

In our experiments, we used two igus® Humanoid Open Platform robots [2]. Each of them is equipped with a dual-core i7-4500U 2.4 GHz processor and a 720p Logitech C905 USB camera. On this hardware, the whole detection, tracking and identification pipeline takes around 50 ms, making it suitable for real-time applications. We performed four different tests to evaluate the proposed system. All tests were conducted on a RoboCup artificial grass field, and the results were manually evaluated for a subset of the frames by the user. The data that was used in the evaluation included varying lighting conditions, and partial, short term and long term occlusions. In the first experiment, we examined the output of the robot detection module by counting the number of successful detections and false positives. The second experiment tested the success rate of the foot detection. A detected position was declared successful if it was within a maximum of 8 pixels from the true bottom pixel of the robot. The third experiment tested the success rate and average error of the visual heading estimation, as compared to the ground truth heading output broadcasted by the corresponding robot. A success was declared if the angular deviation was under 18°, half the size of the heading classes. In the final experiment, the robot identification output was verified by counting the proportion of frames in which the robot labels were correctly assigned. The results are summarized in Table 1. Note that in some of the experiments, we used a camera attached to a laptop, and in other experiments we used a further igus® Humanoid Open Platform. Fig. 5 shows example results of detecting, tracking, identifying, and localizing two robots on the soccer field.

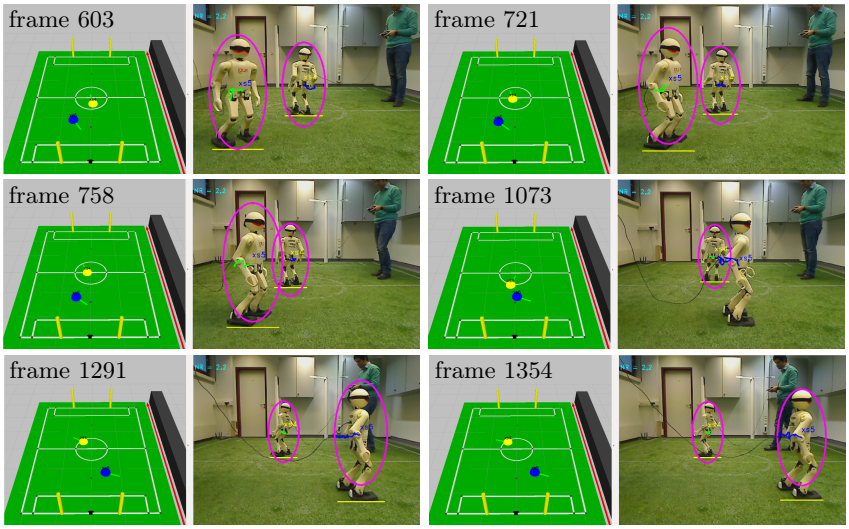

Fig. 5. Detection, tracking, and identification results obtained by our system.

As an extension of the results, we conducted two further experiments where the final robot locations were broadcasted by the observer, and the robots used solely this localization information to walk to a predefined location on the field Fig. 6. A video of the expriment is available at our website[1] The cameras of the robots were covered to demonstrate that they were not using their own visual perception.

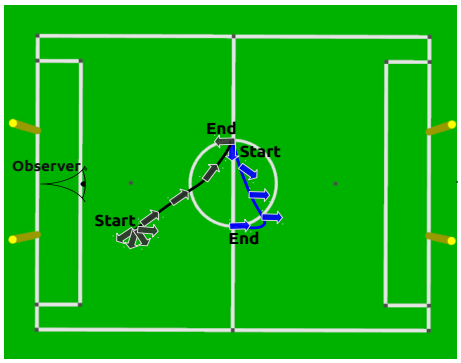

Fig. 6. Positioning experiment with blindfolded robots.

[1] Video link: `https://www.ais.uni-bonn.de/videos/RoboCup_Symposium.2016`

Table 1. Robot detection, heading estimation, and identification results.

Test	Success rate	False positives	Average error	Frames
Robot detection	88%	7	–	1000
Foot detection	89%	–	–	932
Heading estimation	74%	–	17°	845
Robot identification	90%	–	–	932

7 Conclusions

In this paper we proposed a real-time vision pipeline for detecting, tracking, and identifying a set of homogeneous humanoid robots, and gained promising results in experimental verification thereof. Unlike many other works, we could not use any visual robot differences to cope with partial or complete occlusions, so we exploited a heading estimator to identify and track each robot. The result can be used in many RoboCup and real-world scenarios, such as for example shared localization on a soccer field, external robot control, and the monitoring of a group of humanoid robots using a standard camera. As future work, we would like to extend the robot identification to use additional data association cues, such as for example if a robot has fallen down or left the field. Additionally, we would like the observed robots to use their resulting tracked location to improve their own localization.

8 Acknowledgment

This work was partially funded by grant BE 2556/10 of the German Research Foundation (DFG). The authors would like to thank Philipp Allgeuer for help in editing the article and assisting in performing experimental tests.

References

1. J. G. Allen, R. Y. Xu, and J. S. Jin. Object tracking using camshift algorithm and multiple quantized feature spaces. In *Pan-Sydney area workshop*, 2004.
2. P. Allgeuer, H. Farazi, M. Schreiber, and S. Behnke. Child-sized 3D printed igus humanoid open platform. In *Humanoid Robots, IEEE-RAS 15th*, 2015.
3. M. Arenas, J. Ruiz-del Solar, and R. Verschae. Detection of Aibo and humanoid robots using cascades of boosted classifiers. In *RoboCup: Robot Soccer World Cup XI*, pages 449–456. Springer, 2007.
4. C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:1–27, 2011.
5. C. Cortes and V. Vapnik. Support-vector networks. *Machine learning*, 1995.
6. N. Dalal and B. Triggs. Object detection using histograms of oriented gradients. In *Pascal VOC Workshop, ECCV*, 2006.
7. D. F. Dementhon and L. S. Davis. Model-based object pose in 25 lines of code. *International Journal of Computer Vision*, 15(1-2):123–141, 1995.
8. H. Farazi, P. Allgeuer, and S. Behnke. A monocular vision system for playing soccer in low color information environments. In *10th Workshop on Humanoid Soccer Robots, IEEE-RAS Int. Conference on Humanoid Robots*, Korea, 2015.

9. S. Liao, X. Zhu, Z. Lei, L. Zhang, and S. Z. Li. Learning multi-scale block local binary patterns for face recognition. In *Advances in Biometrics*. Springer, 2007.
10. R. Lienhart and J. Maydt. An extended set of haar-like features for rapid object detection. In *International Conference on Image Processing (ICIP)*, 2002.
11. C.-Y. Lin and E. Setiawan. Object orientation recognition based on SIFT and SVM by using stereo camera. In *Robotics and Biomimetics (ROBIO), IEEE International Conference on*, pages 1371–1376, 2008.
12. D. G. Lowe. Object recognition from local scale-invariant features. In *Computer Vision (ICCV), 7th IEEE International Conference on*, pages 1150–1157, 1999.
13. R. Marchant, P. Guerrero, and J. Ruiz-del Solar. Cooperative global tracking using multiple sensors. In *RoboCup: Robot Soccer World Cup XVI*. Springer, 2013.
14. A. Milan, L. Leal-Taixé, K. Schindler, and I. Reid. Joint tracking and segmentation of multiple targets. In *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, pages 5397–5406, 2015.
15. J. Ruiz-Del-Solar, M. Arenas, R. Verschae, and P. Loncomilla. Visual detection of legged robots and its application to robot soccer playing and refereeing. *International Journal of Humanoid Robotics*, 7(04):669–698, 2010.
16. J. Ruiz-del-Solar, R. Verschae, M. Arenas, and P. Loncomilla. Play ball! *IEEE Robot. Automat. Mag.*, 17(4):43–53, 2010.
17. M. Schwarz. NimbRo network library. Github, 2015.
18. S. H. Shaikh, S. Roy, and N. Chaki. Recognition of object orientation from images. In *Emerging Trends in Science, Engineering and Technology (INCOSET), International Conference on*, pages 260–263. IEEE, 2012.
19. Z. Wang, X. Jiang, B. Xu, and K. Hong. An online multi-object tracking approach by adaptive labeling and kalman filter. In *Conference on Research in Adaptive and Convergent Systems*, pages 146–151. ACM, 2015.
20. J. Xing, H. Ai, and S. Lao. Multi-object tracking through occlusions by local tracklets filtering and global tracklets association with detection responses. In *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, 2009.
21. C. Yang, R. Duraiswami, and L. Davis. Fast multiple object tracking via a hierarchical particle filter. In *Computer Vision (ICCV) 10th IEEE International Conference on*, pages 212–219, 2005.
22. Q. Yu, G. Medioni, and I. Cohen. Multiple target tracking using spatio-temporal markov chain monte carlo data association. In *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*. IEEE, 2007.
23. T. Zhang, B. Ghanem, S. Liu, and N. Ahuja. Robust visual tracking via multi-task sparse learning. In *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, pages 2042–2049, 2012.
24. Q. Zhu, M.-C. Yeh, K.-T. Cheng, and S. Avidan. Fast human detection using a cascade of histograms of oriented gradients. In *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, pages 1491–1498, 2006.