The 6th Workshop on Humanoid Soccer Robots @ Humanoids Conference, Bled (Slovenia), October 2011.

Grid-Based Occupancy Mapping and Automatic Gaze Control for Soccer Playing Humanoid Robots

Stefan Kohlbrecher [1], Alexander Stumpf [2], Oskar von Stryk [3]

Simulation, Systems Optimization and Robotics Group, Technische Universität Darmstadt
Hochschulstr. 10, 64289 Darmstadt, Germany
[1]kohlbrecher@sim.tu-darmstadt.de
[2]stumpf@sim.tu-darmstadt.de
[3]stryk@sim.tu-darmstadt.de

***Abstract*—With advances in walking abilities of autonomous soccer playing humanoid robots, the world modeling and state estimation problem moves into focus, as only sufficiently accurate and robust modeling allows to leverage improved locomotion capabilities. A novel approach for dense grid-based obstacle mapping in dynamic environments with an additional application for automatic gaze control is presented in this paper. It is applicable for soccer playing humanoid robots with external sensing limited to human-like vision and strongly limited onboard computing abilities. The proposed approach allows fusion of information from different sources and efficiently provides a single consistent and robust world state estimate despite strong robot hardware limitations.**

I. Introduction

In recent years autonomous humanoid robots have shown significant progress in their locomotion capabilities. However, weight constraints limit the computational power and sensors that can be carried onboard. Additionally, in the RoboCup Humanoid League, only human-like passive external sensors such as vision or hearing are allowed, while active sensors like LIDARs or RGB-D cameras may not be used. The provided sensor data additionally is degraded by significant disturbances during humanoid walking. These strong resource constraints pose significant challenges for state estimation and behavior control and demand a system that employs the limited sensing abilities intelligently.

In this paper, we present an approach that allows autonomous humanoid soccer robots to learn a dynamic grid-based environment map of their surroundings using vision-based perception as well as team communication and prior knowledge about the environment. Furthermore, we present an approach for active gaze control based on this grid map. Unlike previous approaches, we do not only consider static landmarks and a single dynamic object (the ball in robot soccer) for active gaze control. Instead, our approach considers all dynamic and static objects for active gaze control and thus is able to not only improve the performance of self localization and ball modeling but also of obstacle and environment modeling.

II. Related Work

Due to weight restrictions and the accompanying limitations in available computational power as well as limited external sensing abilities and occuring disturbances, world modeling for humanoid soccer robots is more challenging than for many other platforms which face no such limitations.

First introduced as evidence grids in [1] occupancy grid maps are currently the state of the art approach for enabling navigation of mobile robots in real-world environments [2]. The environment is discretized using an evenly spaced grid, with each cell holding a probabilistic estimate of its occupancy. For mobile robots, laser scanners are currently the most widely used sensor for localization and/or mapping. Different methods for estimating this occupancy value exist, commonly used methods are representing the probability of occupancy using a log-odds representation [3] as well as a method commonly dubbed 'counting model' which relies on counting the frequency of a cell seen as free versus the frequency of the cell seen as occupied [4]. The presented approach shares the underlying software for grid map representation with a SLAM approach used within the RoboCup Rescue environment [5]. For the SLAM application, the environment of the robot is mostly considered to be static. Less frequently, occupancy grid maps are used for mapping and state estimation in dynamic environments [6].

For the humanoid soccer scenario, robot centric obstacle models are often used successfully. In the Standard Platform League, grid maps in cartesian [7] or polar coordinates [8] based on the onboard ultrasonic range sensors of the Nao robot are used. These local obstacle models work well for short term planning, but they fail to provide a consistent model of the robot's surroundings that stays valid when the robot moves.

Existing approaches for active gaze control often consume significant computational resources and often are intended to improve robot performance for a specific task like navigation [9], [10]. This limits the feasibility of such approaches for the highly dynamic RoboCup soccer scenario where several cognitive tasks are competing for sensor input and where low onboard computational resources are a limiting factor. Only recently first active gaze control approaches have been published for humanoid soccer Nao robots [11], [12]. However, these consider only self localization and ball tracking problems, but not tracking and detection of obstacles, which play an important role in the adversarial soccer scenario.

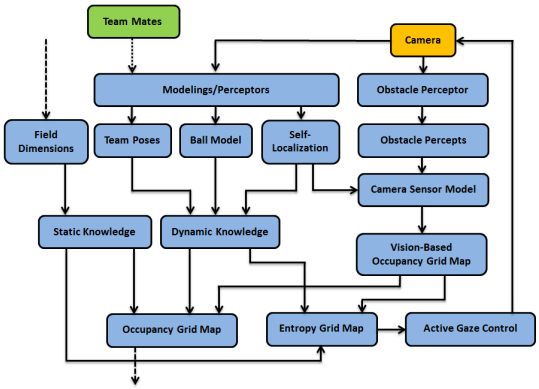


Fig. 1: Data flow and processing steps used by our approach.

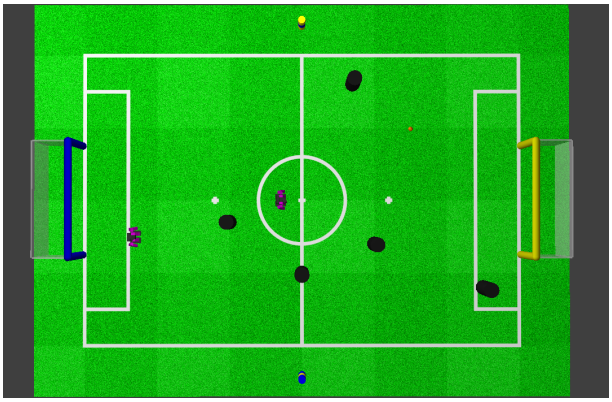


Fig. 2: Top view of the example scenario in simulation

III. Robot Platform and Environment

The autonomous humanoid soccer platform used for this work is based on the HR30 platform by the Hajime research institute. With additions and modifications the current model is named DD2010. It has 21 DOF, which are actuated by Robotis Dynamixel servo motors (18xRX-28, 3xRX-64). Low level and hard real-time control are provided by a microcontroller board with a Renesas SH2 CPU [13], while high level cognition runs on a FitPC2 board mounted on the back of the robot, using an 1.6GHz Atom Z530 CPU. Both systems are connected by a serial (RS232) connection. Among the internal sensors of the robot are a 3 axis accelerometer board as well as 3 single-axis angular rate gyroscopes. The joint servos can be queried to report their measured joint angles. A more comprehensive overview is available online [14].

The environment is defined by the RoboCup Humanoid KidSize League [15]. The playing field has a size of 6x4 meters and consists of green carpet with white field lines. Goals and landmarks are color coded blue and yellow. In a competition, teams of three fully autonomous robots compete against each other.

IV. Overview

The contribution of this paper is twofold: A novel approach for real-time learning of dynamic obstacle maps and an approach for active gaze control based on this learned grid map are introduced. We address both in two distinct Sections. Fig. 1 gives an overview of the data flow and processing steps of the developed system. Numerous heteregenous sources of information are leveraged. Apart from vision-based obstacle detections, also static prior knowledge about the environment and dynamically modeled knowledge about moving objects on the field are used to learn a detailed model of the environment.

The scenario used as an example throughout this work is depicted in Fig. 2. It uses a full 3D kinematic and approximate dynamic simulation. The considered robot is situated near the center circle and oriented towards the yellow goal. Multiple obstacles simulating opponent robots are surrounding it. A teammate robot is situated close to the blue goal. The ball is lying to the left front relative to the robot.

V. Obstacle Model

With the strong restrictions imposed by the humanoid platform and RoboCup rules, many proven methods used in other mobile robots cannot be used. Active external sensors like laser range finders are not allowed and methods like stereo vision imply weight penalties due to sensor size and required onboard processing abilities that prohibit their use. For this reason, we use an obstacle detection approach based on images provided by the monocular camera of the robot. As our method relies on reliable self-localization and obstacle detection by vision, we provide a short overview of our self localization approach and vision system.

A. Self-Localization

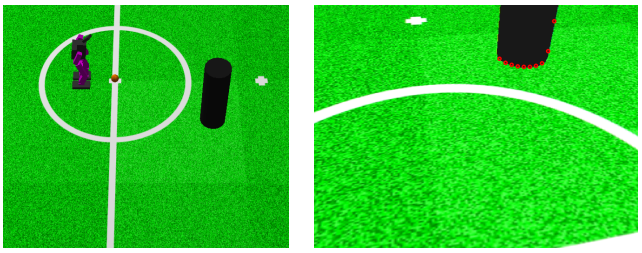
Self Localization is performed based on the commonly used and proven Monte-Carlo-Localization approach [16]. We extend the original approach by using sensor resetting with a preceding particle maturing step before particles are inserted into the sample set. Here, samples generated from observation first have to incorporate observations from a number of different landmarks before they are allowed to be inserted into the particle set used for state estimation. This greatly reduces the amount of faulty pose hypotheses introduced into the sample set and hence improves robustness.

Similar to [17], we also track a unique ancestry identification number for every created particle which gets copied when particles are copied during the resampling step. The particle cluster having the most particles with same ancestry then is selected for estimating the robot pose and a Gaussian approximation of this particle cluster is used to estimate pose uncertainty. Importantly, the eigenvectors and eigenvalues of the covariance matrix of this Gaussian approximation provide information about the principal directions of uncertainty of the robot pose estimate and can be used as a cue as to which viewing directions for the sensors are beneficial to reduce pose uncertainty. Being a compact representation of the estimated robot state, the Gaussian pose estimate can also be easily transmitted via wireless communication and thus be used for coordination in the cooperating team of robots.

B. Vision-Based Obstacle Detection

A color-based system based on scanlines similar to the one described in [18] is used for vision-based object detection. The grid size used for the scanline-based sampling of the image is selected depending on the pose of the camera relative to the playing field. As the ball is the smallest object that has to be detected, the grid size is adapted to enable detection of the ball. This means a coarse grid can be used for areas of the image that are close the robot, while distant areas have to be sampled using a finer grid.

The RoboCup Humanoid KidSize rules specify the allowed color and shape of robots and artificial obstacle poles, so all black objects can be considered to be obstacles in this scenario. For obstacle detection, vertical scanlines start at the bottom of the image and processing proceeds towards the top. For every sample point on the scanline, the corresponding pixel is either determined to be free, unknown or occupied using color classification. If a sufficiently large amount of pixels is determined to be occupied, scanline processing stops and the scanline is marked as occupied. Both the image coordinates as well as the robot coordinates of the start and endpoints for the scanline are written to the obstacle percept, which contains detection results for all processed scanlines after processing the image is finished. Fig. 3 (b) shows a camera image with obstacle points marked by red dots.



(a) Scenario in simulator (b) Camera image

(c) Sensor model

Fig. 3: Illustration of camera sensor model: (a) shows an overview of the scenario. (b) shows the robot camera image with detected obstacle points shown in red. (c) shows the application of the sensor model. The blue arrow depicts the robot pose. The red lines denote the robot field of view projected to the ground and the red dot marks the intersection point of the optical axis of the camera with the ground plane.

C. Learning Dynamic Occupancy Grid Maps

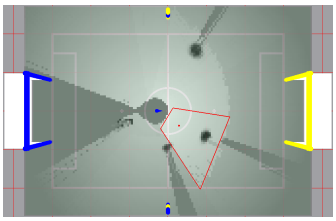
As described in Section II occupancy grid mapping is a well-established method for many applications in mobile robotics. However, to the authors' knowledge, a comprehensive approach fusing vision-based and other information has not been used so far on resource constrained humanoid robots. Unlike existing approaches, we estimate the occupancy grid map in world coordinates and rely on our self-localization system to provide consistent robot pose estimates. Measures to increase the robustness of the self localization system as described in Section V-A make this possible. We thus employ a mapping with known poses approach, taking the pose estimates and their covariance for granted.

1) Grid Representation: The map is represented using a regular grid that discretizes the state space, in this case the playing field, into rectangular cells. For each of these cells, the probability of occupancy is estimated. For performance reasons, our implementation uses a modified counting model representation for the cell state. Here, a count of the number of times the cell was seen as free (c_f) as well as occupied (c_o) is kept per cell and used to approximate the probability of occupancy. To account for the dynamic nature of the environment, the original approach is modified to also keep track of a reliability estimate for the occupancy, resulting in the following estimation of the occupancy probability:

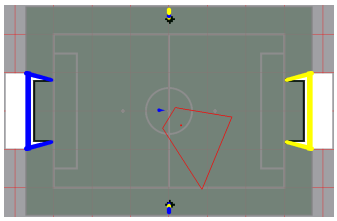
$$p(\mathbf{m}_i) = \begin{cases} 0.5, & \text{if } c_f + c_o = 0 \\ \dfrac{c_o}{c_f + c_o} * r + 0.5 - \dfrac{r}{2}, & \text{otherwise} \end{cases} \quad (1)$$

with r being a reliability estimate that decays with the time the cell has not been observed. Using this approach, occupancy probability can be estimated independently of reliability. If r reaches zero, the grid cell data is reset to the initial estimate, as it has not been observed for an extended period of time.

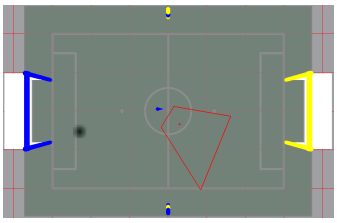
2) Camera Sensor Model: As can be verified in Fig. 3 (c) obstacle perceptions by the monocular camera system as described in Section V-B projected to the ground bear similarity to LIDAR data commonly used in 2D SLAM approaches. While the uncertainties generally are higher, the principle underlying the map update using an inverse sensor model is similar to that commonly employed with LIDAR data [19]. First, the current camera field of view has to be projected onto the map, so the start- and endpoint of every percept scanline can be determined in map coordinates. We then use the Bresenham line algorithm [20] to update all cells between as free. The angular uncertainty is ignored during the update of free space because of the added computational complexity and minor impact on overall accuracy. If scanlines end with an obstacle, the endpoint is updated using the current estimated angular uncertainties, by updating the map with a 2D Gaussian approximation of the endpoint. The approximate projection method for determining the corresponding covariance matrix from the estimated angular uncertainties of the robot sensor system is described in [21] and omitted here for brevity.



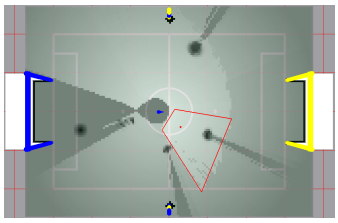
(a) Vision-based occupancy grid map



(b) Static knowledge



(c) Dynamic knowledge



(d) Fused occupancy grid map

Fig. 4: Different sources of obstacle information used: (a) Vision based map. (b) Static obstacle based map (grid cells that belong to landmark poles and goals are occupied). (c) Dynamic obstacle based map (here, the teammate is incorporated). (d) Resulting map fusing the three preceding maps.

3) Other Sources of Obstacle Information: Obstacles detected by the vision system are only one source of information. The location of static obstacles is known beforehand and thus can easily be incorporated into the occupancy grid map. Pose estimates reported by teammates are also considered. Here, the Gaussian approximation of a teammate's current pose estimate provided by its self-localization system are received via the team communication system and added to the obstacle map. The final occupancy grid map thus fuses all available sources of information containing information about obstacles. It can easily be extended for other sources of information about obstacles, for example a dedicated detector for opponent players. Fig. 4 provides an overview of the different types of obstacle information and the fused occupancy grid map.

VI. Active Gaze Control

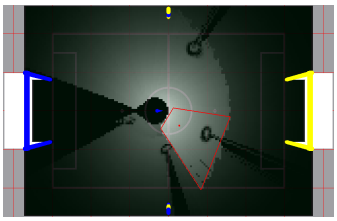
There are several state estimation systems that are competing for information provided by the camera system of the robot. Self localization relies on acquisition of known static landmarks like goal poles, field lines or field line intersections. Ball modeling requires tracking of the most dynamic object on the field, the ball. Obstacle modeling on the other hand requires regularly scanning the area of the field close to the robot in all direction as well as looking into the direction of already known obstacles.

The occupancy grid map provides information about dynamic and static objects that are relevant for obstacle avoidance for the robot. Known static objects as defined by the field layout and setup of the RoboCup Humanoid KidSize League are relevant for self-localization. The ball model provides a Gaussian state estimate of the ball, incorporating position and velocity as well as the accompanying covariance matrix.

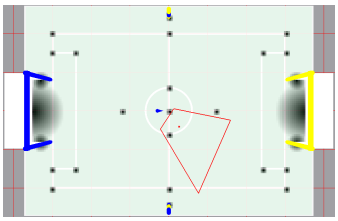
Fusing these sources of information, an entropy map that provides the base for an entropy based active gaze control approach can be computed. For the vision based occupancy grid map, Shannon entropy can be determined based on the binary estimate of the probability of occupancy that is available for each cell

$$H(\mathbf{m}_i) = -\left(p(\mathbf{m}_i) log_2 p(\mathbf{m}_i) + (1 - p(\mathbf{m}_i)) log_2 (1 - p(\mathbf{m}_i))\right) \quad (2)$$

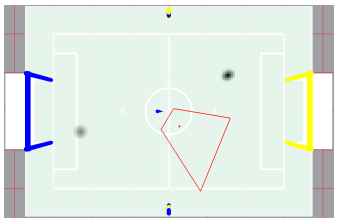
where m_i is the grid cell with index i. Using this equation for every grid cell of the occupancy grid map an approximate entropy map can be created. Next, static landmarks used for self-localization and the ball state estimate are merged to this map. By using different factors during this merging step, prioritization can take place. Among the considered variables for this prioritization are the robot pose and ball state estimates as well as the current role of the robot. Fig. 5 shows an overview of the different sources used and the fused entropy grid map.



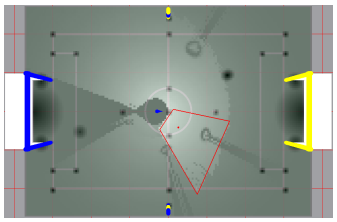
(a) Entropy of occupancy grid map



(b) Entropy of static knowledge



(c) Entropy of dynamic knowledge



(d) Fused entropy grid map

Fig. 5: The learned vision-based occupancy grid map is transformed into the entropy representation in (a). Static (b) and dynamic knowledge (c) are added into the entropy grid map. The resulting fused entropy grid map is shown in (d).

4) Candidate Score: As an exhaustive search for the best gaze direction for every cognition cycle is computationally not feasible on the resource constrained humanoid robot platform, we employ a sampling based approach. A list of candidate gaze directions is kept during the update phase of the entropy map. These candidate gaze directions are represented by the intersection point of the optical axis of the camera with the ground plane. For every update of a cell c_i of the entropy grid map, first a check is performed if c_i is inside the area of feasible gaze directions. If this test passes, a weight depending on the position of the cell in the feasible area is computed:

$$w_{space}(c_i) \quad := \quad w_{space,d}(c_i) \cdot w_{space,\beta}(c_i), \quad (3)$$

with $w_{space,d}(c_i)$ being a weight for the distance of c_i from the robot and $w_{space,\beta}$ being a weight for the angle from the robot to c_i around the vertical axis. The current head pose is weighted using a similar approach:

$$w_{head}(c_i) \quad := \quad w_{head,\alpha}(c_i) \cdot w_{head,\beta}(c_i), \tag{4}$$

with $w_{head,\alpha}(c_i)$ being a weight depending on the change in angle the head pitch servo has to perform and $w_{head,\beta}(c_i)$ being a weight depending on the change in angle the head yaw servo has to perform to point the optical axis at c_i. For both $w_{space}(c_i)$ and $w_{head}(c_i)$ we currently employ simple triangular functions that have their maximum when the needed angular change amounts to zero. The total weight is given by:

$$w(c_i) \quad := \quad w_{space}(c_i) \cdot w_{head}(c_i) \tag{5}$$

Based on this weight and the sampled entropy value at the cell position $H(c_i)$, two candidate points c_a and c_b can be compared:

$$p_{c_a} < p_{c_b} := w(c_a) \cdot H(c_a) < w(c_b) \cdot H(c_b) \tag{6}$$

Given the order defined by this comparison operator, the candidate point is inserted into the candidate point list if the distance between it and the already existing points in the list is above a threshold value. Using this approach, redundant locations in the gaze candidate point list are surpressed. To keep computation time low, a small list is used in practice, keeping only the highest scoring candidate points.

5) Field of View Entropy: The list of candidate points as described previously is based only on the entropy value of a single grid cell. Therefore, it remains to be shown how an approximate information gain for the given candidate points can be computed. For every one of the candidate points, the approximate field of view entropy (FOVE) is determined by fitting a Gaussian into the projection of the corresponding camera field of view projected to the ground. The Gaussian approximation leads to higher weights for objects closer to the camera center, which is desirable as partial visibility of objects near the border of camera images can cause detection failures in image processing. For each candidate point at position $(x, y)^T$ the mean field of view entropy is determined by:

$$\overline{H_{FOVE}(x, y)} = \frac{\sum_{x_i} \sum_{y_i} (1 - \Delta(x_i, y_i)) \cdot H(c_i)}{n}, \tag{7}$$

where $(x_i, y_i)^T$ takes the position of every cell within the approximated FOV and n denotes the amount of cells considered in the weighted entropy sum. $\Delta(x_i, y_i)$ describes the Mahalanobis distance relative to the fitted Gaussian. As the mean field of view entropy only has to be computed for a small number of candidate points, real-time computation is feasible. The candidate point scoring the highest mean field of view entropy is selected as the next gaze point and handed off to the camera control system.

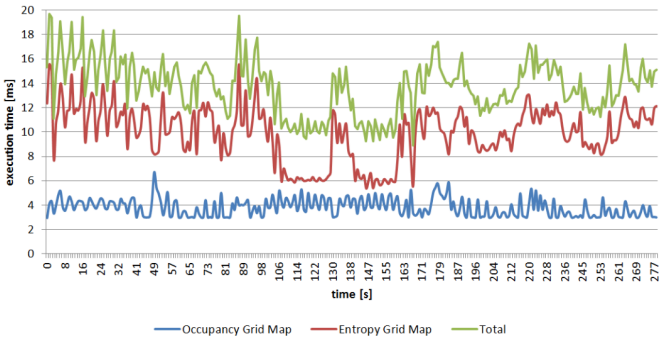


Fig. 6: Timing plot for obstacle mapping (blue), entropy map estimation (red) and the sum of both (green).

VII. Evaluation

Multiple factors can be considered for the performance evaluation of the system. Among those are the runtime efficiency as well as a comparison to a previously employed approach. A comprehensive test scenario employing the obstacle map for path planning cannot be shown here, as this is the subject of current work. All tests during evaluation have been performed using real robot hardware.

A. Runtime Performance

The approach has to run in real-time on the restricted onboard hardware of humanoid robots. It has to allow for processing of every image acquired by the camera, so reaction time of the system is as fast as hardware permits. This means that the whole cognition loop has to run at a rate of 30Hz. We performed several measurements using the onboard 1.6GHz Atom CPU. As can be seen in Fig.6 the update rate is kept when using a grid map of 5cm resolution and using obstacle mapping and active gaze control. The occupancy grid map is updated in a mean time of 3.8 ms with little standard deviation, while entropy map estimation takes between 5.4 and 15.6 ms. In total, a runtime of 20 ms is never exceeded, which is sufficient to keep the total cognition cycle time below the 33 ms that are needed to process every image on our system. The data was recorded using a simulated match situation with the robot first taking the goalie role and then switching to the striker role at the 160 seconds mark.

B. Obstacle Mapping Performance

Using a simple robot centric obstacle model for comparison, it can be shown that the presented approach exhibits less error and models obstacles in the environment with higher accuracy. For evaluation, a single obstacle pole was placed in the field of view of the robot at different distances. Every 500ms the distance of all estimated obstacle positions to the real obstacle position was computed. The standard deviation of these samples was then taken as a measure of obstacle modeling accuracy. As can be seen from Table I, the proposed approach outperforms the previously used sector-based approach in all experiments, exhibiting a lower standard deviation of the error for the obstacle position estimate. A

video showing the qualitative performance of the system on a robot moving towards the ball in the presence of obstacles is available online[1]. To minimize bandwidth requirements, the grid cell state is represented by the three discrete options free, occupied and unknown here.

	experiment 1	**experiment 2**	**experiment 3**
σ sector model	227.92 mm	227.2 mm	383.0 mm
σ obstacle grid map	33.0 mm	50.69 mm	192.55 mm

TABLE I: Standard deviation of obstacle model accuracy.

C. Active Gaze Control Performance

Using a manual camera control system similar to those used in other soccer robots as a baseline, it can be shown that the proposed approach for active gaze control enables better coverage of the environment of the robot while at the same time considering all important perception tasks. In the experiment, the robot moved a predefined trajectory, while looking at the ball situated at the center circle. The "look at ball" head behavior is compared to the active gaze control approach in Table II. A video of the system used for active gaze control on real robot hardware is available online[2].

experiment 2	**active gaze control**	**look at ball**
ball model validity [%]	79.96	97.01
soccer field observed [%]	75.31	53.33
robot's front observed [%]	100.00	45.02

TABLE II: Observation coverage when the robot moves around the soccer field.

VIII. Conclusion

A novel approach for obstacle mapping and active gaze control is presented in this paper. Inspired by well established approaches for solving the SLAM problem, a real-time grid based obstacle map of the environment is learned online during operation of a severly resource constrained humanoid robot using only onboard sensors and processing. Additionally, an active gaze control system accounting for the competing cognitive tasks of ball tracking, self localization and obstacle detection is presented, which also runs in real-time.

Directions of future work are the use of the presented results for realtime robot path planning and cooperative behavior. Also, feeding back information about the planned path to the active gaze control system as well as exchange of maps between robots will be investigated.

Acknowledgment

The authors would like to thank the team members of the Darmstadt Dribblers for their support.

[1]http://www.youtube.com/watch?v=NeAUSwqmUhM

[2]http://www.youtube.com/watch?v=IL1hSfpxpsw

References

[1] H. Moravec and A. Elfes, "High resolution maps from wide angle sonar," in *Proc. IEEE Intl. Conf. on Robotics and Automation*, 1985, pp. 116–121.

[2] E. Marder-Eppstein, E. Berger, T. Foote, B. Gerkey, and K. Konolige, "The office marathon: Robust navigation in an indoor office environment," in *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, 2010, pp. 300 –307.

[3] S. Thrun, W. Burgard, and D. Fox, "Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)," 2005.

[4] D. Hähnel, R. Triebel, W. Burgard, and S. Thrun, "Map building with mobile robots in dynamic environments," in *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, 2003.

[5] S. Kohlbrecher, J. Meyer, O. von Stryk, and U. Klingauf, "A flexible and scalable slam system with full 3d motion estimation," in *International Symposium on Safety, Security, and Rescue Robotics*. IEEE, November 2011.

[6] C. Coué, C. Pradalier, C. Laugier, T. Fraichard, and P. Bessière, "Bayesian occupancy filtering for multitarget tracking: an automotive application," *The International Journal of Robotics Research*, vol. 25, no. 1, p. 19, 2006.

[7] T. Röfer, T. Laue, J. Müller, A. Burchardt, E. Damrose, A. Fabisch, F. Feldpausch, K. Gillmann, C. Graf, T. J. de Haas, A. Härtl, D. Honsel, P. Kastner, T. Kastner, B. Markowsky, M. Mester, J. Peter, O. J. L. Riemann, M. Ring, W. Sauerland, A. Schreck, I. Sieverdingbeck, F. Wenk, and J.-H. Worch, "B-human team report and code release 2010," 2010, only available online: http://www.b-human.de/file_download/33/bhuman10_coderelease.pdf.

[8] E. Chatzilaris, E. O. I. Kyranou, A. Paraschos, E. Vazaios, N. Spanoudakis, N. Vlassis, and M. G. Lagoudakis, "Kouretes 2010 spl team description paper," 2010, only online: http://www.tzi.de/spl/pub/Website/Teams2010/Kouretes.pdf.

[9] J. F. Seara, K. H. Strobl, and G. Schmidt, "Path-Dependent Gaze Control for Obstacle Avoidance in Vision Guided Humanoid Walking," in *Proc. IEEE Intl. Conf. on Robotics and Automation*, Taipei, Taiwan, September 2003, pp. 887–892.

[10] W. Burgard, D. Fox, and S. Thrun, "Active mobile robot localization by entropy minimization," in *Proc. of the Second Euromicro Workshop on Advanced Mobile Robots*. IEEE Computer Society Press, 1997.

[11] A. Seekircher, T. Laue, and T. Röfer, "Entropy-based Active Vision for a Humanoid Soccer Robot." RoboCup International Symposium 2010, 2010.

[12] S. Czarnetzki, S. Kerner, and M. Kruse, "Real-time Active Vision by Entropy Minimization Applied to Localization." RoboCup International Symposium 2010, 2010.

[13] D. Scholz, M. Friedmann, and O. von Stryk, "Fast, robust and versatile humanoid robot locomotion with minimal sensor input," in *Proc. 4th Workshop on Humanoid Soccer Robots / IEEE-RAS Int. Conf. on Humanoid Robots*, Paris, Dec. 7-10 2009.

[14] Darmstadt Dribblers. (2010) www.dribblers.de. [Online]. Available: http://www.dribblers.de/

[15] RoboCup Soccer Humanoid League Rules and Setup, "www.tzi.de/humanoid," Tech. Rep., 2010.

[16] F. Dellaert, D. Fox, W. Burgard, and S. Thrun, "Monte carlo localization for mobile robots," in *IEEE Intl. Conf. on Robotics and Automation (ICRA99)*, May 1999.

[17] T. Laue and T. Röfer, "Pose extraction from sample sets in robot self-localization - a comparison and a novel approach," in *Proc. 4th European Conf. on Mobile Robots*, I. Petrović and A. J. Lilienthal, Eds., Mlini/Dubrovnik, Croatia, 2009, pp. 283–288.

[18] T. Röfer et al., "GermanTeam 2005," HU-Berlin, U-Bremen, TU-Darmstadt, U-Dortmund, Tech. Rep., 2005.

[19] S. Thrun, "Learning occupancy grids with forward sensor models," *Autonomous Robots*, vol. 15, pp. 111–127, 2002.

[20] J. Bresenham, "Algorithm for computer control of a digital plotter," *IBM Systems Journal*, vol. 4, no. 1, pp. 25–30, 1965.

[21] S. Kohlbrecher and O. von Stryk, "Modeling observation uncertainty for soccer playing humanoid robots," in *Proc. 5th Workshop on Humanoid Soccer Robots at the 2010 IEEE-RAS Int. Conf. on Humanoid Robots*, Nashville, Dec. 6-8 2010.