Distributed Sensing in a Robotic Soccer Team

Stefan Niemczyk, Dominik Kirchner, Andreas Witsch, Stephan Opfer, Kurt Geihs*

Distributed Systems, University of Kassel,

Wilhelmshöher Allee 73

Kassel, Germany

Email: {niemczyk, kirchner, witsch, opfer}@vs.uni-kassel.de

*Email: geihs@uni-kassel.de

Abstract—In robot teams a coherent world model is essential for cooperative behavior. The world model is a collection of information about the robots environment. In order to establish a coherent world model sensor fusion is required. However, sensor noise injects uncertainty on the observations and can lead to conflicting behavior. In this paper, we present a world model for multi-robot systems. Our approach distinguishes between a local and a shared world model. The latter achieves consensus by redundant computations of shared sensor information. We evaluated our approach in an appropriate environment, i.e., the RoboCup Middle-Size League.

I. INTRODUCTION

The coordination of autonomous mobile robots can inherently benefit from precise coherent shared data of the environment. However, sensor noise and communication delay inhibits fast consensus on shared data. In practice, sensor fusion is one of the most common methods, to reduce the sensor noise and extract the most valuable information from shared data [4]. The performance of sensor fusion approaches is often evaluated in scenarios with a team of cooperative robots.

An example scenario for a cooperative robot team is RoboCup¹. In particular, the RoboCup Middle-Size League (MSL) is characterized by unpredictable opponent behaviours and high velocities, i.e., 5 m/s on a $18 \text{ m} \times 12 \text{ m}$ large field.

In MSL, the perception is primarily done by an omnidirectional vision system. The range limitations of current omni-directional cameras, forbid players to observe the whole field on their own. Sensor fusion of all available and useful information within the team helps to achieve reliable and coordinated game play. This is facilitated by the distinction between local and shared world models as shown in [5].

In this paper, we present our experiences with distributed sensing based on our MSL-team *Carpe Noctem Cassel* (CNC) [2]. In general, we perform all computations redundantly on all robots and assume missing information to be unknown. For example, if a robot does not receive ball observations of other robots, it exclusively relies on its own observations. This approach is robust against broken robots and unreliable communication, as information of other robots is not mandatory for local reasoning. Furthermore, our approach leads to a more precise and complete world model the more information of other robots are available.

The rest of the paper is organized as follows. Section II describes our distributed sensing used in RoboCup. In Section III, we discuss the strengths of our approach and highlight further extensions. We conclude the paper and provide ideas for future work in Section IV.

II. DISTRIBUTED SENSING

This section describes the local information processing of a CNC robot and the sensor fusion used to improve these data based on the shared team information. We use the obstacle detection as an example to clarify the described concepts. In order to specify the information sharing, we start with a short description of the CNC world model.

A. World Model

The CNC world model contains all information the robot collects about its environment. These information are computed based on sensor information and can be used by all decision processes. Furthermore, the world model is split into a local part (referred to as local world model) and a shared world model. The local world model contains the information inferred from local sensor information, e.g., the position of the robot on the field, provided by its own localization routines. The shared world model contains the information that are shared within the team of robots, e.g., the position of other robots on the field. All positions are represented in a global reference frame with the point of origin in the centre of the field.

While the local information processing is performed with 30 Hz, information is only shared with 10 Hz to reduce the used bandwidth. This information sharing is not synchronized within the team. As a result, the timestamps of the information can vary up to 100 ms, ignoring network latencies. Sensor fusion of this unsynchronized information needs to deal with these variations. However, sending all information at the same time would cause unnecessary transfer collisions.

B. Local Sensing

This section gives a short introduction to the local sensing of the CNC robot. Obstacle and ball detection are based on the camera image, which has a resolution of 640×480 pixels and is recorded with 30 Hz. Therefore, all processing must be performed in less than 33 ms to retain the robot's reactivity. Figure 1(a) shows the omnidirectional vision system, where the camera looks at a convex mirror. The mirror enables the system to capture a 360° view, as depicted in Figure 1(b). However, the detection range is limited, because of the reduced $\frac{px}{arrea}$ -ratio at the border of the image.

¹http://www.robocup.org/





(a) Camera and mirror

(b) Omnidirectional image

Fig. 1. Figure (a) shows the camera aligned on the convex mirror and Figure (b) shows a recorded omnidirectional image.

All robots are required to be black. This allows for an efficient obstacle detection, by operating on a gray image. We utilize scan lines for further performance improvement. Scan lines are radially distributed with an interval of 6° and start from the center of the image. Changes in brightness along a scan line, induced by transitions between the lighter field and the black robots are used to detect obstacles. The nearest matching pixels on the scan lines create a distance profile, centred around the robot.

The next step is an adaptive discretisation of the distance profile. Therefore, similar distances in the neighbourhood are combined to a single value. In the last step the obstacles need to be divided into multiple robots based on variations of robot sizes.

Similar to the obstacle detection, the ball detection is designed to determine the position of a ball with a given colour, see the orange balls on Figure 1(b). First, the algorithm identifies regions of interest (ROI), by finding colours similar to ball colours. Again, scan lines are used to find matching colours and similar results in the neighbourhood are combined to a single ROI.

Based on these ROIs, a template matching is used to identify ball hypotheses. Therefore, an edge detection inside the ROIs is performed on the original gray image. The normal vectors of a circular object are pointing to or away from the centre of the circle and can be matched by a corresponding patterns. Our implementation requires at least 12 pixel to identify a ball hypothesis. This leads to a theoretical detection range of 11 m. In practice, the robots are able to detect the ball up to 7 m away. Additionally, we represent uncertainty of ball hypotheses with a mean and a covariance matrix.

However, the achievable precision of the hypothesis is limited by the local sensor values. Furthermore, teammates and opponents can not be distinguished by local sensor values only.

C. Sensor Fusion

This section describes the information sharing and fusion within the team. To counteract the limited range of local sensing, environment information is shared within the team via Wifi connection. Obviously, this leads to duplicated information of identical objects. We apply sensor fusion techniques to merge this information in order to create a global world view and to reduce uncertainty about the environment. However, a poor realization of sensor fusion will lead to wrong estimations and a worse team play. Therefore, the algorithms need to deal with uncertain, imprecise, unreliable, and possibly outdated information. Moreover, efficiency is a further important aspect, because every time new local information arrives, the fusion with shared information needs to be performed. Extending the example from the last section, we describe our realization of merging obstacles and ball estimations in the RoboCup domain.

The initial situation of the obstacle fusion is an unordered set of shared obstacle positions and teammate positions. As a clustering algorithm, we choose a hierarchical clustering. Due to the dynamic environment, the number of physical obstacles on the field is unknown. The algorithm merges clusters until a given threshold is exceeded, so the total number of clusters can be unknown. We select a method based on Ward [7] to decide when to stop the clustering. Ward uses the inner variance of clusters before and after merging. The variance $TD^2(C)$ for a given cluster C is given by:

$$TD^{2}(C) = \sum_{p \in C} dist(p, \mu_{C})^{2}$$
(1)

where p is a point from cluster C and μ_C is the cluster centroid. The function dist denotes the Euclidean distance. A smaller value of $TD^2(C)$ means that the points are closer to each other [1]. The variance difference for two given cluster P and Q can then be computed by:

$$dist_{var}(P,Q) = TD^{2}(P \cup Q) - (TD^{2}(P) + TD^{2}(Q))$$
 (2)

where $P \cup Q$ is the merged cluster of P and Q. Afterwards, a decision is made whether a cluster represents an opponent or a teammate. Therefore, the shared position of the teammates were added to the list of obstacles at the beginning. Each cluster that contains a position of a teammate is annotated accordingly. This cluster will be removed from the following steps and the shared position of the teammate itself is used. To avoid conflicts, no cluster can contain more than one teammate.

To deal with delayed or noisy information, the clustering provides higher priority for own data about nearby obstacles. For all clusters within a given radius (e.g. 1.5 m) the detected obstacle position by the robot itself is used instead of the cluster centroid (see Figure 2).

Furthermore, we need to remove false positives from the clustered positions. False positives are clusters that do not represent real obstacles and result from noisy sensor information. To identify false positives, all opponent clusters are annotated with the number of robots s that see the obstacle and the number of robots d that do not see the obstacle, but should see it. The line-of-sight criteria is checked to determine, if a robot could see an obstacle. A majority vote decides if a cluster is valid or not. If s < d the cluster will be removed and if $s \ge d$ the cluster will be kept.

Figure 2(a) shows two robots and the sensed positions of four numbered obstacles. The first robot Rob1 (blue) sees two obstacles (1 and 2). The second robot Rob2 (yellow) sees four obstacles (1 to 4). After applying the above described



Fig. 2. Figure (a) shows the local obstacles of *Rob1* (blue) and *Rob2* (yellow). Figure (b) shows the resulting fused obstacles (red).

algorithm, the observations are grouped into four clusters. The red circles in Figure 2(b) are the resulting positions. Both robots can see obstacle 1 (s = 2) and the resulting position is the cluster centroid. Obstacle 2 is also seen by both robots. However, the obstacle is near *Rob1* (*dist* < 1.5 m) and the resulting position is the locally sensed position from *Rob1* instead of the cluster centroid. Clusters 3 and 4 are only seen by *Rob2* (s = 1). The line-of-sight from *Rob1* to obstacle 3 is interrupted by obstacle 2, so d = 0 and with s > d the cluster is kept. The line-of-sight from *Rob1* to obstacle 4 is not interrupted and so the robot should see it (d = 1). However, in case of a tie the obstacle is kept.

Afterwards, an object tracking is applied to refine the results and derive further information, like opponent velocities. A detailed description of the tracking is given in [3].

Contrary to the obstacle detection, the fusion of ball observations from multiple robots is based on Dempster-Shafer Theory of Evidence (DST) [6]. Presenting the whole approach is beyond the scope of this paper and hence only a short overview is given. A detailed description can be found in [4]. The algorithm starts with the ball hypotheses \mathcal{H} based on the shared information. Each hypothesis is annotated with the involved robots.

The next step is to compute for all hypotheses of each robot the believe mass m where $m : H \rightarrow [0, 1]$. Afterwards, the hypotheses are combined by applying the Dempster-Shafer Rule of Combination in order to fuse multiple observations:

$$m(x) = \sum_{H \in \mathcal{H}} m(x|H) \cdot m(H), \tag{3}$$

where m is the belief mass for a given ball position x considering all ball hypotheses \mathcal{H} . Hypotheses are represented with a mean and covariance matrix and the mass assignment of Equation 3 results in a Gaussian Mixture Model (GMM). The combination of two GMMs is reduced to a point-wise multiplication. In order to deal with network latency, the resulting believe mass for a given hypothesis is discounted by:

$$\alpha = 1 - e^{-\frac{1}{t_s} \cdot (t_{mr} - t_0)},\tag{4}$$

where t_0 is the timestamp of the hypothesis which is discounted and t_{mr} is the timestamp of the most recent observation. t_s specify the time in which the position could

have changed significantly (in case of the ball 50 ms), which results in no intersection of two successive observations. The combination believe mass of two hypothesis H_i and H_j is computed by:

$$m_{1,2}(H_{ij}) = \alpha \cdot A^*_{H_i \oplus H_i} \cdot m_1(H_i) \cdot m_2(H_j), \qquad (5)$$

where $A_{H_i \oplus H_j}^*$ describes the agreement of the two hypothesis H_i and H_j [4]. Thus, the ball hypothesis with the highest resulting mass is selected as ball.

III. DISCUSSION

The presented distributed sensing approach addresses the requirements of a coherent, global world view by sharing information within the robot team. This shared information compensates the limited sensor range of a single robot and enables global team coordination. In order to improve the accuracy of the shared information, we apply sensor fusion techniques tailored for the RoboCup domain. These techniques merge duplicated, noisy information to an improved estimation of the world state. Additionally, we specify the uncertainty of these estimations. This allows the robots to make decisions with respect to the degree of uncertainty. As a result, a precise and efficient team coordination is achieved.

In addition to these advantages, we would also like to mention some shortcomings of our current information sharing approach. Our robots periodically share their local information every 100 ms, regardless of the existence of new information. This leads to an unnecessary communication overhead. Thus, our approach does not scale for large team sizes and high amounts of shared information. In order to enable large teams a flexible information sharing is needed.

Moreover, the used world models are currently domainspecific and hard coded in a monolithic fashion. This results in an inflexible system and changes to the software may lead to unpredictable side effects. Additionally, an adaptive reconfiguration of the world model at runtime is not possible. Furthermore, in the current implementation it is not possible to share information only within subgroups. Although, these features are not required in RoboCup, a more flexible world model may be required to transfer the approach to other application domains.

IV. CONCLUSION AND FUTURE WORK

A global world view based on local sensor readings is a challenging requirement for multi-robot domains. In this paper we presented our experiences in distributed sensing in the RoboCup domain. We extend the local sensing range by sharing local information within the team. In order to further improve the accuracy of the shared information, we apply sensor fusion techniques, like the Dempster-Shafer Theory of Evidence. Additionally, we combine sensor fusion with domain-specific heuristics to eliminate false positives. This combination has proven its quality to implement enhanced team coordination as required in the RoboCup Middle Size League. However, while this solution works fine in the addressed domain, we recognize limitations and expect necessary modifications for other application domains. Our future research aims at providing a more flexible and scalable world modelling approach. This would enables a simple description and reconfiguration of a local and a shared world model. Furthermore, we are currently developing a generic approach to resolve conflicts for optimistically shared information in adverse environments. This is a crucial requirement to achieve mutual agreement on common goals in multi-robot teams.

REFERENCES

- [1] M. Ester and J. Sander. *Knowledge Discovery in Databases: Techniken und Anwendungen.* Springer Berlin Heidelberg, 1 edition, 2000.
- [2] T. Haque, D. Kirchner, S. Opfer, D. Saur, A. Witsch, K. Geihs, and et. al. Team description: Carpe noctem 2013. Technical report, 2013.
- [3] S. Opfer, H. Skubch, and K. Geihs. Cooperative path planning for multi-robot systems in dynamic domains. In *Mobile Robots - Control Architectures, Bio-Interfacing, Navigation, Multi Robot Motion Planning* and Operator Training. InTech, 2011.
- [4] R. Reichle. Information Exchange and Fusion in Dynamic and Heterogeneous Distributed Environments. PhD thesis, University of Kassel, Distributed Systems Group, 2010.
- [5] M. Roth, D. Vail, and M. Veloso. A real-time world model for multi-robot teams with high-latency communication. In *International Conference of Intelligent Robots and Systems*, 2003.
- [6] G. Shafer. A Mathematical Theory of Evidence. Princeton University Press, Princeton, 1976.
- [7] J. Ward Jr. Hierarchical grouping to optimize an objective function. *Journal of the American statistical association*, 58(301), 1963.