A Software Architecture for Distributed Visual Tracking in a Global Vision Localization System*

Siniša Šegvić and Slobodan Ribarić

University of Zagreb,
Faculty of Electrical Engineering and Computing,
Unska 3, 10000 Zagreb, Croatia,
sinisa.segvic@fer.hr,
http://www.zemris.fer.hr/~ssegvic/index.html

Abstract. The paper considers detecting and tracking multiple moving objects in real time by a multiagent active vision system. The main objective of the envisioned system is to maintain an explicit comprehensive representation of the scene by combining individual views obtained from several autonomous observer nodes. In order to allow for a near real time performance, each observer agent has been assigned a separate computer system, while an additional coordination agent is introduced for relieving the observers from correspondence and coordination tasks. The proposed architecture is specially suited for monitoring ground objects whose dimensions are relatively small when compared to the dimensions of the scene. This assumption makes it possible to speculate a ground object 3D position from the single view, which consequently allows a robust correspondence approach. The architecture has been implemented in an experimental global vision system, whose final objective is to provide localization information for a group of simple robots without vision sensors. The system was preliminary tested in the laboratory environment, and the obtained experimental results are presented.

1 Introduction

Visual tracking is an important field of computer vision, in which the movement of the objects in the scene is inferred from the acquired sequence of image frames. Many applications in this field deal with complex 3D scenes, in which approaches based on single point of view face considerable limitations. The problems include limited coverage of complex scenes, speculative and imprecise 3D tracking results, and various occlusion ambiguities and failures. Some of these limitations may be alleviated by using active vision [1] or panoramic sensors [2], but the most robust solution can be achieved only in a distributed vision system, by combining evidence obtained from several adequately placed observer nodes.

Most of the existing distributed visual tracking (DVT) systems focus on tracking humans in indoor environments [3–5, 2, 6]. These designs have been

* This work has been supported by the Croatian Ministry of Science and Technology, Contract Number 2001-072.

motivated either by surveillance [4–6] or general human-computer interaction [3, 2] applications. An another important application field of DVT is the real time monitoring of various sport events. The information about the game status can be used for augmenting a broadcast TV edition by an overlay image showing the positions of the players or the ball [7], which are difficult to estimate from the current view. Additionally, the obtained data could be employed for a semi-automated direction of the TV edition. In such an arrangement, the viewing directions of all cameras covering the scene might be adjusted by the automated control system, in order to achieve an acceptable presentation of the event.

The proposed work has been directly inspired by a yet another application of DVT, and that is providing localization information to a group of simple autonomous mobile robots with modest equipment (see fig.1). This approach has been called global vision [8, 9], distributed vision [10, 1] and sensor network for mobile robotics [11], and has been classified with artificial landmark localization techniques [12], since it requires special interventions to the environment in which the navigation takes place. The approach is particularly suitable for applications requiring a large number of autonomous vehicles (e.g. an automated warehouse), because it allows trading fixed cost vision infrastructure for a per-vehicle savings in advanced sensor accessories [8]. Recently, global vision has become a popular method for coordinating "players" in small robot soccer teams (see e.g. [9]).

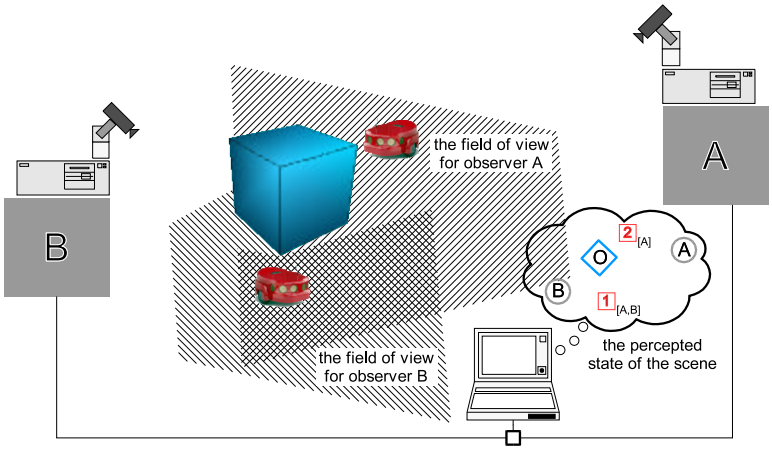


Fig. 1. Overview of a global vision localization system.

In most realistic global vision applications, it is feasible and favourable to place the cameras above the navigation area so that the objects appear relatively small in images acquired from each viewpoint (see fig.1). The proposed architecture therefore assumes that the position of each tracked ground object can be estimated from a single view. In order to improve the tracking quality and simplify the implementation of the overall control, it is advantageous to consider

autonomous observers, capable of adjusting the viewing direction according to the movement of the tracked objects. Consequently, the observers are organized in a multiagent system [13], in which some of the actions are taken autonomously while others are done in coordination with other observers.

The following section gives a brief overview of the previous work in related research directions. The proposed multiagent architecture is outlined in section 3, while sections 4 and 5 provide some of the implementation details for the two types of agents within the system. Experimental results are shown in section 6, while section 7 contains a short discussion and directions for the future work.

2 Previous work

Previous researchers identified many useful design patterns and ideas for building DVT systems. Multiple viewpoints have been employed because they allow: disambiguating occlusions [6]; monitoring structured scenes (e.g. corridors) [3, 4]; determining exact 3D position of the tracked object [7, 1]; solving difficulties tied to the limited field of view [10, 6, 5, 1]; fault tolerance [2]. In order to ensure the flexibility and openness of the system, it has been assumed that the observers are not mutually synchronized and that they have different processing performance [3, 2, 1]. Consequently, a special protocol has been needed to synchronize the clock of each observer to the referent time. Active vision [1] and panoramic cameras [2] have been used in order to enlarge the visible portion of the scene from each viewpoint. The most significant DVT architectures are outlined in the following list, in the decreasing order of centralization:

1. Monolithic system: raw images from all cameras are processed within the same program, there is no per-view autonomous processing [4, 11].
2. Hierarchical division of responsibility: each observer node is assigned a dedicated computer system, while the observations are gathered and processed in a centralized fashion within a higher level program [2, 6].
3. Decentralized common view: observers communicate the tracking results to all peers, so that each observer stores a copy of the common view [3].
4. Society of independent watchers: observers independently localize the objects within the visible portion of the navigation area; the tracking is performed within per-object agents by combining evidence from relevant observers [10].
5. Society of cooperative agents: each observer tracks a single object and adjusts the viewing direction accordingly [1]; observers dynamically form groups tracking a common object, and may be unaware of other objects' movement.

Different architectures suite different configurations with respect to the parameters such as count of observers n_{obs}, count of tracked objects n_{to} and whether active cameras are available [1]. In general, the decentralized approaches are more flexible with respect to scalability and fault tolerance. However, the intelligent behaviour of the system tends to be more complicated to express through control protocol between peer components, than within a single component of the hierarchical structure. Thus, architectures 4) and 5) have been

employed in systems for which $n_{obs} \gg n_{to}$, in which suboptimal resource allocation is affordable. The architecture proposed in this paper aims at many realistic applications for which $n_{obs} \leq n_{to}$ and consequently combines the effectiveness of the hierarchical structure with the flexibility of autonomous observers.

Multiagent organization [13] has recently become an often proposed software architecture paradigm. Building systems in terms of intelligent anthropomorphic components is appropriate when the communication between the large parts of a system becomes complex enough, so that it becomes useful to model it after the human interaction. A good description of the multiagent paradigm has been articulated as the agenthood test [14], which states that a system containing one or more reputed agents should change substantively if another reputed agent is introduced to the system. The test stresses that the operation of a multiagent system depends on mutual awareness of its components.

3 The proposed architecture

According to fig.1, the desired system should possess the following capabilities:

- tracking objects of interest within a single observer by an active camera;
- integration of the data obtained from several observers to the common scene representation, by assuming different observer performances;
- coordination of the viewing directions of the observers for the purpose of achieving a good tracking of the state in the scene;
- robustness with respect to the removal of existing or adding new observers;
- soft real time performance.

The architecture design is mostly determined by the requirement that a computer vision algorithm is required to operate in the real time environment. Because of the complexity of vision algorithms, it is favourable to ensure that each observer agent gets most of the time of a dedicated processor, and to assign data integration and coordination tasks to the coordinator agent running on a separate computer. The resulting architecture is outlined in fig.2: observers send measurements to the coordinator, which integrates the data into the common view and controls the observers behaviour in an opportunistic manner.

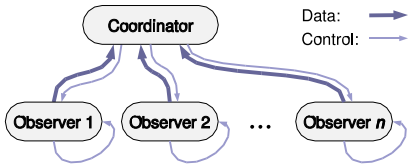


Fig. 2. The top level view of the multiagent architecture.

The system organized after fig.2 satisfies the agenthood test cited in the Introduction, if the coordinator is viewed as a part of communication infrastructure.

Whenever a new observer registers with the coordinator, the responsibilities of all observers are rescheduled in order to obtain a better coverage of the scene.

4 Implementation details for the observer agents

Observer agents are responsible for detection and tracking of objects of interest, as well as for adjusting the viewing direction of the associated camera with the purpose of following the current object or searching for new objects. The desired system consists of several observers so that, besides coordinate systems of the image (o, x, y) and the camera (C, X, Y, Z), it is necessary to define the common referent coordinate system of the scene (O, K, L, M). An important property of considered scenes is that the objects of interest move within the horizontal ground plane π. It is therefore convenient to align the pan axis of the camera with the normal of π, and to choose the camera and the world coordinate systems for which the upright axes Z and M coincide with that direction (see fig.3).

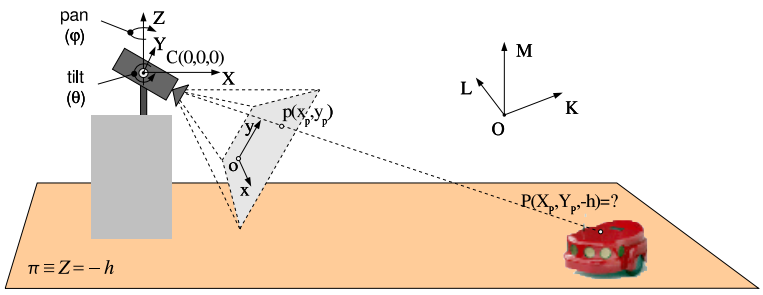


Fig. 3. The observer agent imaging geometry.

In order to speculate the 3D position in camera coordinates $P(X_P, Y_P, -h)$ from the position of the object in the image plane $p(x_p, y_p)$, it is necessary to perform several transformations, based on precalibrated intrinsic and extrinsic [15] camera parameters and the known angular position of the camera (ϕ, θ). In theory, the only error of the obtained position is caused by the finite height of the tracked object, but in practice several other errors come into effect. These errors are due to imperfect estimations of camera parameters and compensations of lens distortions and geometric inadequacies of the camera controller (offset of the projection center from the crossing of pan and tilt axes).

The main requirement for observer agents is the real time detection and tracking of objects of interest within the current field of view. Additionally, they are required to exchange the following data with the coordinator: (i) clock synchronization and extrinsic camera parameters (at the registration time), (ii) the current viewing direction (after each change), and (iii) the time stamped list of

detected objects in camera coordinates (after each processed image). Observers operate in one of the following modes with respect to autonomous camera movement: seeking (camera seeks for an object and then the mode is switched to 'tracking'), tracking (viewing direction follows the active object), or immobile (viewing direction does not change). Finally, they listen for control messages from the coordinator and switch operating modes or move the camera accordingly.

5 Implementation details for the coordinator

The basic responsibilities of the coordinator encompass integration and analysis of individual object positions reported by the observers. The integration task sums up to repetitive updating of the common representation of the scene. The data structure holding the representation is the central component of the coordinator, and is organized in four hierarchical levels with gradual increase of abstraction: (i) object positions (measurements), (ii) individual trajectories, (iii) top level objects, and (iv), trajectories of top-level objects (see fig.4).

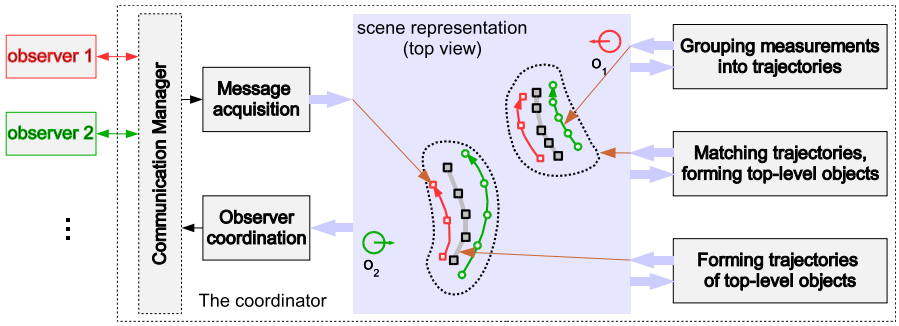


Fig. 4. Block diagram of the coordinator agent.

5.1 Overall architecture

The five basic procedures of the coordinator agent are (see fig.4): message acquisition, grouping measurements into trajectories (temporal integration), matching trajectories into top-level objects (spatial integration), formation of top-level trajectories, and observer coordination. These procedures transform the lower level structure components into the higher level ones, and their activation order depends on run-time detected conditions, such as the arrival of a new measurement, or when a certain observer looses the tracked object from its visual field. The required opportunistic activation can be adequately expressed within the blackboard [16, 17] design pattern, which is often used in the distributed solving of the complex problems. The main subjects in such organization are knowledge

sources (the basic procedures), the central data structure or blackboard (common view of the scene) and the control component which triggers the activation of knowledge sources (see fig.5).

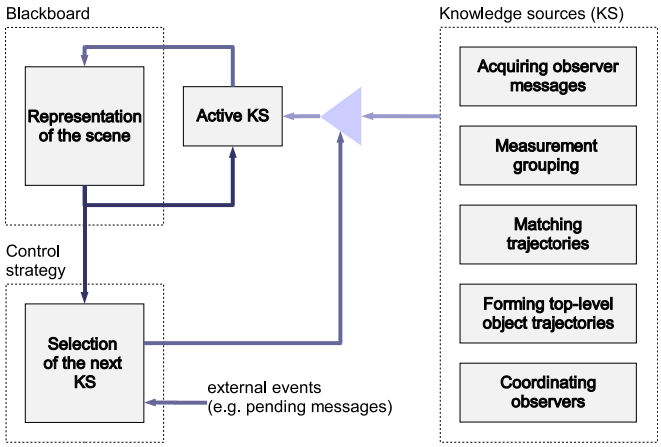


Fig. 5. The proposed coordinator architecture.

5.2 Matching individual trajectories

An individual trajectory is a temporal sequence of measurements reported by a certain observer, for which it is believed that they correspond to the same object. Each measurement contains the object position converted to world coordinates, as well as the acquisition time of the image in which the object was detected. The matching procedure establishes correspondence between recent segments of trajectories containing measurements obtained within the last two seconds. Each of the obtained correspondence sets defines a 3D position of the top level blackboard object which should correspond to a real object in the scene. The correspondence procedure is different from clustering because, during the procedure, some trajectories become mutually incompatible and can not be grouped together. This occurs whenever the trajectories belong to the two correspondence sets both of which contain trajectories reported by the same observer.

The main difficulty in matching a pair of recent trajectory segments reported by different observers is caused by the assumption that the observers are not synchronized, i.e. that single measurements in corresponding trajectories have different acquisition times. The problem has been solved by (i) finding the time interval for which both segments are defined, (ii) taking N equidistant time instants within that interval, and (iii) interpolating representative points in both trajectories within that instants. The procedure is illustrated for N=5 in fig.6, where the synchronized representative points are designated with crosses.

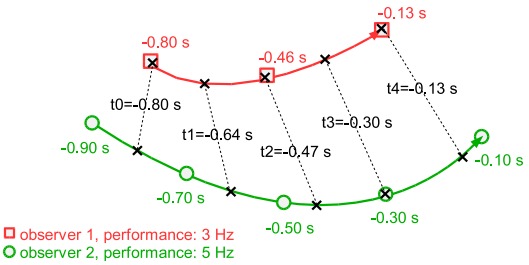


Fig. 6. Finding synchronized sets of representative points in two trajectory segments.

As described in section 4, measurements of the same object recorded by different observers may systematically differ due to multiple sources of error. Experiments have shown that the dominant effect of these errors to short trajectory segments can be modeled as a simple translation. The matching is therefore based on a distance function computed as a weighted sum of mean value and standard deviation of the displacements between the corresponding representative points. The correspondence procedure follows a greedy iterative approach, such that in each iteration the least distant pair of matchable trajectories is associated together until the distance is greater than a predetermined threshold.

5.3 Coordination of the observers

In general, the coordination of the observers is a complex task since it is composed of at least the following two contradictory requirements: (i) precise position determination for each tracked object, and (ii), monitoring the empty parts of the scene for appearance of new objects. The optimal coordination strategy is necessarily application specific, since it depends on many parameters such as the counts of observers and objects of interest, the priority of individual objects, whether all observers can "see" the entire scene, etc. The following terms may prove useful in the design of a strategy:

- an object tracked by exactly one observer is defined as *weak*;
- an observer tracking a weak object is defined as *bound*;
- an observer tracking an object which is tracked by exactly one additional observer is defined as *important*;
- an observer tracking 0 objects is defined as *idle*;
- an observer which is neither important nor bound nor idle, is defined as *free*.

It seems that the real time performance will remain the most challenging requirement for quite some time, so that the communication protocol between the coordinator and the observers should not prescribe waiting for confirmation messages. This can be achieved by scheduling the activation of the coordination procedure in regular time intervals (e.g. 2 seconds). In such an arrangement, the procedure in each invocation examines the situation on the blackboard, issues

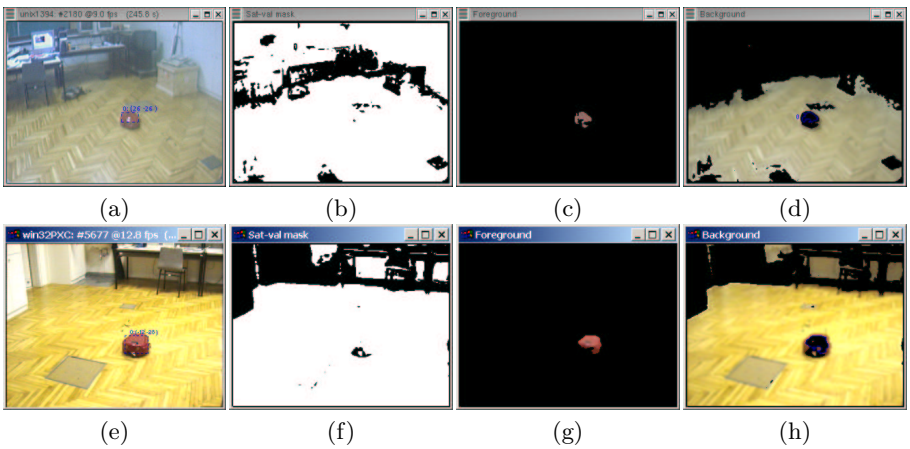
one or more control messages to the observers, and returns to the blackboard control component (no answer from the observer is required). The following minimalistic strategy has been devised for robust (although suboptimal) coordination in partially occluded scenes containing a small number of objects.

1. observers in seeking operating mode, bound, important, free and idle observers are assigned priorities of 0, 0, 1, 2 and 3, respectively;
2. if there are no observers with a non-zero priority, no action is performed;
3. otherwise, the highest ranked observer (round robin scheme is used to choose among observers with the same priority) is chosen and is denoted as O_C;
4. if there is a weak object A positioned outside the field of view of O_C (otherwise, A is occluded from O_C), O_C is assigned the tracking of A;
5. otherwise, only if O_C is not important, it is switched to the seeking mode.

6 Experimental results

The experimental system was tested in a heterogeneous environment, with three observers running under different operating systems connected to the Ethernet LAN. Individual applications within the system (the agent program, the observer program, calibration and testing utilities) were built from the version control system managed library containing about 50000 lines of C++ source code.

Fig. 7 shows experimental results for the two observers tracking the same object. In the experimental implementation, the objects are detected on the basis of their colour and the colour of the surrounding background. For each observer, the figure shows the original image with the designated detected object (a,e), the saturation-value mask used for eliminating regions that are too dark or too light (b,f), and connected regions which are, according to the hue of the corresponding pixels, classified as objects (c,g) or background (d,h).



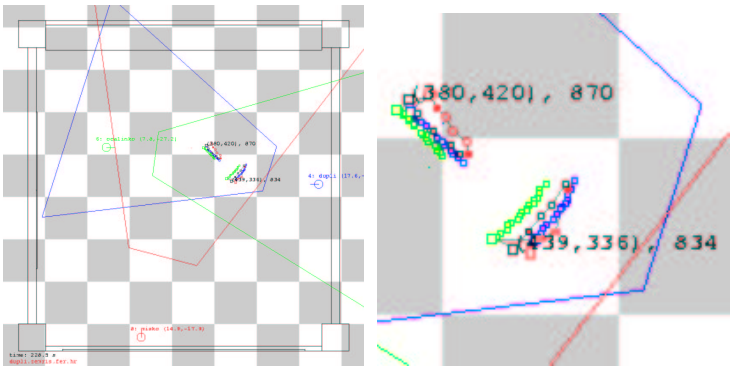
(a) (b) (c) (d)

(e) (f) (g) (h)

Fig. 7. Experimental results for observers A (a-d) and B (e-h); see text for details.

One of the observers was running on a multiprocessor computer which made it possible to run the coordinator on the same computer without a performance hit. Simple but effective procedures for object detection and tracking allowed for high observer performance of 12.5 Hz and 9.1 Hz on computers with approximate single processor SPEC CINT2000 base performances of 710 and 530, respectively.

Fig. 8 shows the top view of the scene with two objects moving at speeds of about 0.5 m/s, which is computed in real time within the coordinator agent. The figure background contains the referent one-metre grid and the walls of the lab in which the experiment takes place. Each of the three registered observers is designated with the circle indicating the observer position, the short line showing its orientation, and the polygonal area designating the respective fields of view. Finally, the detected objects are designated with their last positions and recent trajectory segments (as reported by observers), as well as with positions, approximate areas and trajectories of the respective top level blackboard objects.



(a) (b)

Fig. 8. The top-level view of the scene with three observers and two tracked objects (a), and the enlarged central portion in which the detected objects are situated (b).

7 Conclusions and the future work

A hierarchical multiagent DVT architecture suitable for a large class of realistic problems has been proposed. The behaviour of the described coordination procedure is comparable to the recent solution proposed in [1], but the hierarchical coordination approach has a great potential for more sophisticated behaviours due to the availability of the explicit common view. Eventual network congestion problems arising for large observer counts could be overcome by extending the architecture with a "recursive" coordinator type, being able to perform as an observer responsible to the coordinator agent at a higher hierarchical level.

The obtained experimental results confirmed that the proposed architecture is a viable approach for putting together the required software components in

a manageable, flexible and extensible manner. The future work will be directed towards refinements to the existing architecture in order to achieve more involved coordination schemes, as well as towards dealing with procedures for diminishing the systematic error in observer measurements, and ensuring robustness of the system in the view of the physical contact of tracked objects.

References

1. Matsuyama, T., Ukita, N.: Real-time multitarget tracking by a cooperative distributed vision system. Proceedings of the IEEE **90** (2002) 1136–1150
2. Karuppiah, D., Zhu, Z., Shenoy, P., Riseman, E.: A fault-tolerant distributed vision system architecture for object tracking in a smart room. In: Proceedings of the International Workshop on Computer Vision Systems, Vancouver, Canada (2001) 201–219
3. Nakazawa, A., Kato, H., Inokuchi, S.: Human tracking using distributed vision systems. In: Proceedings of the International Conference on Pattern Recognition. Volume I., Brisbane, Australia, IEEE (1998) 593–596
4. Cai, Q., Aggarwal, J.: Tracking human motion in structured environments using a distributed-camera system. IEEE Transactions on Pattern recognition and Machine Intelligence **21** (1999) 1241–1247
5. Khan, S., Javed, O., Rasheed, Z., Shah, M.: Human tracking in multiple cameras. In: Proceedings of the International Conference on Computer Vision, Vancouver, Canada, IEEE (2001) I: 331–336
6. Dockstader, S.L.; Tekalp, A.: Multiple camera tracking of interacting and occluded human motion. Proceedings of the IEEE **89** (2001) 1441–1455
7. Guéziec, A.: Tracking pitches for broadcast television. Computer **35** (2002) 38–43
8. Kay, M., Luo, R.: Global vision for intelligent AGVs. SME Journal of Vision **9** (1993)
9. Veloso, M., Stone, P., Han, K., Achim, S.: The CMUnited-97 small robot team. In Kitano, H., ed.: RoboCup-97: Robot Soccer World Cup I. Springer Verlag, Berlin (1998) 242–256
10. Sogo, T., Ishiguro, H., Ishida, T.: Mobile robot navigation by distributed vision agents. In Nakashima, H., Zhang, C., eds.: Approaches to Intelligent Agents. Springer-Verlag, Berlin (1999) 96–111
11. Hoover, A., Olsen, B.D.: Sensor network perception for mobile robotics. In: Proceedings of the International Conference on Robotics and Automation. Volume I., San Francisco, California, IEEE (2000) 342–348
12. Borenstein, J., Everett, H.R., Feng, L.: Navigating Mobile Robots: Sensors and Techniques. A. K. Peters, Ltd., Wellesley, MA (1996)
13. Wooldridge, M.: Intelligent agents. In Weiss, G., ed.: Multiagent Systems. MIT Press (1999) 27–79
14. Huhns, M.N., Singh, M.P.: The agent test. Internet Computing **38** (1997) 78–79
15. Mohr, R., Triggs, B.: Projective geometry for image analysis. A tutorial given at the International Symposium of Photogrammetry and Remote Sensing, Vienna (1996)
16. Pfleger, K., Hayes-Roth, B.: An introduction to blackboard-style systems organization. Technical Report KSL-98-03, Stanford University, California (1998)
17. Huhns, M.N., Stephens, L.M.: Multiagent systems and societies of agents. In Weiss, G., ed.: Multiagent Systems. MIT Press (1999) 79–121