Abstract

Distributed networks of sensors have been recognized to be a powerful tool for developing fully automated systems that monitor environments and human activities. Nevertheless, problems such as active control of heterogeneous sensors for high-level scene interpretation and mission execution are open. This paper presents the authors’ ongoing research about design and implementation of a distributed heterogeneous sensor network that includes static cameras and multi-sensor mobile robots. The system is intended to provide robot-assisted monitoring and surveillance of large environments. The proposed solution exploits a distributed control architecture to enable the network to autonomously accomplish general-purpose and complex monitoring tasks. The nodes can both act with some degree of autonomy and cooperate with each other. The paper describes the concepts underlying the designed system architecture and presents the results obtained working on its components, including some simulations performed in a realistic scenario to validate the distributed target tracking algorithm.

1. Introduction

The development of environment and activity monitoring systems, based on heterogenous networks of sensors, constitutes an active investigation field, with many potential applications, including safety, security, ambient intelligence and health-care assistance. In real scenarios, such as buildings, airports, road and rail networks, or sport grounds, a single sensor is not able to monitor the whole environment or to track a moving object or a person for a long period of time, due to field of view limitations. Furthermore, integrating information from multiple sensors is a basic requirement for achieving an adequate level of robustness and scalability. Typical technological solutions include the use of camera networks that are able to cooperate to monitor wide areas and track objects beyond the capabilities of each single sensor. The problem of multi-target tracking in distributed camera networks has been addressed in several works. For instance in [12], Kalman-Consensus filter [9] is used to fuse in a distributed fashion the information coming from each camera of the network. Experimental results show that consensus significantly increases the system performance. In [11] an extension of [12] to wide-area scene understanding is presented. To optimize the dynamic scene analysis, a control framework for a PTZ camera network is introduced. In [13], a survey of distributed multi-camera systems for target tracking on planar surfaces is provided. In [14], a review of distributed algorithms for several computer vision applications is presented, and emphasizing the advantages of distributed approaches with respect to centralized systems. As a basic principle, in distributed estimation, each node of the network locally estimates the state of a dynamical process using information provided by its local sensor and by a subset of nodes of the network, called neighbors [16]. In literature, there are several approaches to distributed estimation in sensor networks. Their peculiarity is the presence of an agreement step to minimize the discrepancy among sensory nodes [9, 3, 2].

While the use of multiple sensors increases reliability and effectiveness in large environments, it poses problems related to the need of infrastructures that can be heavy and expensive. These infrastructures can be reduced by exploiting the flexibility of moving sensors mounted on semi or fully autonomous vehicles that can be employed as individual agents or organized in teams to provide intelligent distributed monitoring of broad areas. Mobile sensors may significantly expand the potential of assistive and surveillance technologies beyond the traditional passive role of event detection and alarm triggering from a static point of view. Mobile robots can actively interact with the environment, with humans or with other robots to accomplish more complex cooperative actions [1, 15]. Nevertheless, mobile surveillance devices based on autonomous vehicles are still in their initial stage of development and many issues are currently under investigation [4, 5, 8].
In this paper, a distributed architecture exploiting fixed and mobile heterogeneous sensors to intelligently monitor environments and track human activities is presented. The proposed cooperative monitoring system integrates fixed calibrated cameras with a team of autonomous mobile robots equipped with different sensors. The system is being developed as part of a project, called BAITAH, aimed at identifying and developing ICT-based Ambient Intelligence technologies to support the independent living of fragile people in their domestic environments. In this project, mobile sensors are intended to provide two main contributions: they can provide information about the observed human target in areas that are out of the field of view of fixed cameras (thus reducing complexity and costs of the required infrastructure) and they can move close to the target to increase precision and reliability of scene analysis whenever fixed sensors are unable to provide robust estimates. In designing such a system, a major challenge is the integration of high-level decision-making issues with primitive simple behaviors for different operative scenarios. This aim requires a modular and reconfigurable system, capable of simultaneously addressing low-level reactive control, general purpose and monitoring tasks and high-level control algorithms in a distributed fashion. This paper presents an overview of the system architecture and of the implemented algorithms. Specifically, an extension of the algorithm introduced in [10], namely the Consensus-based Distributed Target Tracking (CDTT) algorithm, is described and validated through simulations in a realistic environment.

2. System Overview

In this section, the overall system architecture is introduced, and the functionalities of the network are described.

2.1. Architecture

A multi agent\(^1\) system for distributed target tracking and monitoring of an indoor environment is being developed. It uses fixed agents, consisting of static cameras, to execute surveillance tasks in areas of relevant interest and a network of mobile agents that are able to perform local and specific monitoring tasks to completely cover the environment. A schematic representation of the system is shown in Fig. 1. Integration among the various agents is performed via a distributed control architecture, which uses a wireless network for communication. In this network, each agent corresponds to an independent software component that is executed on the robots embedded PC for the mobile agents and on a workstation for the fixed cameras. This architectural solution provides several advantages. First, the system is totally plug-&-play, i.e., to increase the number of sensors it is sufficient to add a new camera (with its IP address) to the network, so that no further effort to program the cameras is required. Software maintenance is easy and immediate, avoiding the broadcast updating of each camera software. Moreover, algorithms for motion detection and shadow removal, as will be explained in the next sections, are based on the evaluation of pixel correlation that requires a very fast processing unit to run in real time and that cannot be done efficiently with embedded cameras. All agents can be logically considered as belonging to a peer-to-peer network. They differ only for their own sensor capabilities. In particular, every agent is able to detect an event (e.g., to perceive moving people or objects) and to localize an event (e.g., tracking the position of a person) in the environment using one or more sensor devices, whereas, in addition, mobile agents are able to execute tasks, through their actuators.

2.2. Fixed Nodes

Each fixed node consists of a cameras located in a fixed point of the environment. On the fixed nodes, the following functionalities are implemented [6]:

- **Task1: Motion Detection.** The binary shape of moving objects (e.g., people) is extracted. Specifically, a statistical background model is generated by evaluating mean value and standard deviation for each point. Then, foreground moving regions are detected by estimating, for each pixel, the similarity between the current frame and the background model.

- **Task2: Shadow Removal.** This task is necessary because foreground pixels may correspond not only to real moving objects, but also to their shadows. The shadow pixels need to be removed, because they alter the real shape of objects and decrease the precision of their localization. A connectivity analysis is, finally, performed to aggregate pixels belonging to the same moving object.

- **Task3: Object Tracking.** The detected moving objects, after shadow removal, are tracked over time. Statistical

\(^1\)Hereinafter, the network nodes will be also called agents in order to emphasize their detection, communication and computation capabilities.
(tracked object life time) and spatial information are extracted for each of them. This task enables the association of each moving region to the corresponding target object, based on its appearance. Furthermore, it reduces false detections due to noise or light reflections.

- **Task4: 3D Moving Object Localization.** The intersection of the central axis of the rectangular bounding box containing the moving region with its lower side provides the estimate of object position on the ground plane. The corresponding 3D position is evaluated using a pre-calibrated homographic matrix between the image plane and the 3D ground plane.

### 2.3. Mobile Nodes

The mobile nodes of the network consist of autonomous mobile robots. Each mobile agent is equipped with sensory devices to interact with the environment. Every node must be able to localize itself in the environment, to safely navigate avoiding static and dynamic obstacles, to identify and track the position of a target in the environment. The Robot Operating System (ROS) has been adopted as a framework for communication management, sensor acquisition and actuator control on the mobile robots. It is an open source framework that presents several packages ready to run in order to control all the devices of a robotic platform. ROS provides a Navigation Stack, which enables the robot to navigate in a known environment avoiding obstacles, as well as sensor management packages [7]. The most important peculiarity of ROS is its modular structure that makes it possible to modify or substitute some modules. In order to develop a customized monitoring architecture, new functionalities have been developed and added to the native ROS framework. Specifically, the structure of the navigation stack of ROS has been modified in order to add surveillance capabilities to the mobile nodes. A coordinate transformation from local to global coordinates was also introduced for the people tracking task.

### 2.4. Distributed Target Tracking Algorithm

To perform people tracking using a distributed network of sensors, the monitoring system adopts the fully distributed Consensus-based Distributed Target Tracking (CDTT) algorithm, previously proposed by the authors in [10], extended to mobile sensor networks. It consists of a two-phase iterative procedure: an estimation stage and a consensus stage. In the estimation step, each network node estimates the position of the target. If the node is able to perform a measurement, then it will estimate the target position through the measurement, improved by a Kalman filter. Otherwise, the node will predict the target motion according to the embedded linear motion model of the Kalman filter. In the consensus phase, the individual estimates made by each node converge to a common value through a suitable consensus strategy. Specifically, the agreement among the network nodes is achieved via a customized max-consensus protocol, performed on a measurement accuracy metrics called perception confidence value. This approach was proved to provide good performance in heterogeneous sensor networks composed by nodes with limited sensing capabilities. The CDTT approach is totally distributed, as it does not involve any form of centralization. Moreover, it guarantees that, at each iteration, all nodes have the same estimate of the target position. The reader is referred to [10] for further information.

### 3. Implementation

In this section, details about the implementation of the system are provided. First, the general setup of the sensor network in the experimental environment is introduced, then the hardware characteristics of fixed and mobile nodes are presented.

#### 3.1. Environment Setup

A typical setup used for experimentation of the system is shown in Fig. 2. The picture shows the map of a corridor of the ISSIA-CNR Mobile Robotics and Vision Laboratory, as it is built by the gmapping node available in ROS using the laser data acquired by one of the mobile robots during a complete exploration of the environment. The position of three fixed cameras (C1, C2, C3) is overlaid on the map. Mobile agents (R1) are able to localize themselves in the environment and, using their on-board sensors, they are able to carry out surveillance tasks, such as people detection and tracking. Cameras are calibrated, therefore events detected in the image plane can be located in the real world and their
positions can be communicated to the mobile agents. Mobile robots can explore areas that are unobservable by the fixed cameras and improve the accuracy of events detection by reaching proper positions in the environment. Hence, the proposed system could be useful to reduce the number of fixed sensors or to monitor areas (e.g., cluttered environments) in which the field of view of fixed cameras can be temporarily and dynamically reduced.

3.2. Setup of Fixed and Mobile Nodes

The fixed nodes are three wireless IP cameras ($C_1$, $C_2$, $C_3$) with different spatial resolution, located in different points of the environment (see map in Fig. 2). $C_2$ and $C_3$ are Axis IP color cameras with a $640 \times 480$ pixel resolution and an acquisition frame rate of 10 frames per second. $C_1$ is a Mpixel Axis IP color camera with $1280 \times 1024$ pixel resolution and full frame acquisition rate of 8 frames per second. A calibration step to estimate intrinsic and extrinsic parameters was performed for each camera using the Matlab Calibration Toolbox\(^2\), so that camera coordinates can be mapped to the global world reference frame provided by the map built by the mobile robots. The fixed agent software runs on a workstation linked to each camera by the network infrastructure. The schematic representation of interconnections among nodes composing the Fixed Node module is shown in Fig. 3. For each connected camera an autonomous thread integrated in the ROS framework is implemented, to execute some well-defined ordered tasks as explained in Section 2.2.

The mobile agent ($R_1$ in Fig. 2) consists of a PeopleBot mobile robot platform equipped with a laser range-finder, a Kinect, and an on-board laptop (see Fig. 5). The SICK laser is connected with the embedded robot control unit. The Kinect camera and the PeopleBot control unit are connected with the laptop, via an USB cable and a crossover cable, respectively. The laser range-finder is used to build a map of the environment and to localize the vehicle. The Kinect is used for both navigation (e.g., obstacle avoidance) and high-level tasks such as people detection and tracking. In Fig. 4 a schematic representation of the mobile node module is reported. All ROS nodes run on the on-board laptop, except for sicktoolbox\_wrapper and p2os\_driver. As can be seen, the Navigation Stack of ROS produces robot position estimates, as well as information about obstacles on the basis of laser measurements. The ROS node motion\_control, implemented by our research team, sends velocity references to p2os\_driver ROS node, responsible of the robot guidance. The people\_tracker node estimates the relative position of people with respect to the robot, on the basis of the skeleton information received from openni\_tracker. The relative coordinates of detected people, transformed in the world reference frame, provide input data to the distributed target tracking algorithm.

4. Results

In this section, results concerning the use of CDTT in a simulated environment are presented. First, the simulation setup is described. Then, the results of simulation are shown.

4.1. Simulation Setup

In order to test the CDTT algorithm performance in a realistic scenario, the map shown in Fig. 2 is used as the simulation environment. A target moving inside this map according to various random trajectories is simulated for the target tracking task. Heterogeneity in the sensor network is

\(^2\)The toolbox is available on http://www.vision.caltech.edu/bouguetj/calib\_doc/index.html
due to the different sensing ranges of sensors, set on the basis of the real characteristics of each real device. Specifically, the sensing area is defined as a circular sector area placed at the front of the sensor with radius of \( r_{C_1} = 10 \) meters for camera \( C_1 \), \( r_{C_2} = 8.5 \) meters for camera \( C_2 \), \( r_{C_3} = 7 \) meters for camera \( C_3 \), and \( r_{R_1} = 5 \) meters for robot \( R_1 \). The sensors are modelled as range-bearing, with measurement error depending on distance and bearing of the target relative to the sensor. In order to assess the system performance, attention is focused on the tracking accuracy, by evaluating discrepancy between estimated and actual target trajectory. Specifically, as a metric for target tracking accuracy, the mean square error (in norm) is computed as:

\[
\text{MSE} = \frac{1}{k_f} \sum_{k=1}^{k_f} \| \xi_i(k) - \xi(k) \|^2
\]  

(1)

where \( k \) is the simulated discrete time, \( k_f \) is the duration (in time samples) of the target trajectory, \( \xi_i(k) \) is the actual target position at time \( k \), and \( \xi(k) \) is the global target position estimates performed by the \( i \)-th sensor of the network. It should be noted that the estimated target position is the same for any node of the network, since after convergence of the consensus step of the CDTT algorithm all the network nodes share the same information about the target location.

### 4.2. Simulation Results

The performance of the CDTT is analysed by running a campaign of Monte Carlo simulations in two different cases, i.e., sensor network with four static nodes, and sensor network with three static nodes and one mobile node. Each node is able to survey a given portion of the environment. In particular, it should be noted that although, in general, the mobile robot can move in the whole environment, for this simulation, it is assumed that it can operate in a limited area in the surroundings of its initial location.

A set of 250 random target trajectories is run both for the case of static nodes only and for the scenario including the mobile node. The CDTT simulation for one of the trajectories is shown in Fig. 6. It refers to the simulation performed using a mobile node (green arrow) in addition to the static ones (black arrows). A solid red line denotes the
actual target trajectory, while the estimated positions at each time step $k$ are marked by blue circles. Initially (Fig. 6(a)), the target is sensed by the static node $C_2$, while the other nodes are aware of the target position thanks to the consensus convergence. As soon as the target enters the area surveyed by the mobile node (Fig. 6(b)), the latter approaches the target to perform a more accurate measure.

The numerical results of the simulation campaign are reported in Table 1, showing a mean square error of 0.2339 m and 0.1523 m, for the static network and for the network including the mobile node, respectively. As can be noted, the presence of a mobile node increases the tracking accuracy. This is mainly due to two reasons: first, the mobile node can approach the target, so that it can measure the position of the target with higher accuracy according to the adopted range-bearing sensor model. In addition, the mobile node can track the target also in areas hidden to the fixed nodes, thus increasing the overall coverage of the network.

5. Conclusions

In this paper a novel activity monitoring architecture has been introduced. The main contribution is the combination of fixed and mobile nodes in the monitoring network: mobile sensors enable the complete coverage of large environments with fewer fixed sensors and increase the accuracy of measurements by reaching the most favorable position to observe the current target. The global logical architecture used by the system has been presented. The software agents developed to work on fixed and mobile nodes have been described. Simulations of the behaviour of the system in a realistic environment (with sensor parameters closely corresponding to the characteristics of the real fixed and mobile sensors) have been done and the results have been shown. They have been obtained using a distributed target tracking algorithm developed by some of the authors. The results support the effectiveness of the proposed system and provide a tool to coordinate the proper positioning of mobile nodes in the environment. Future work will compare these results with the data obtained by the real sensors in our lab environment.

Acknowledgements

This research is supported by the National Research Program PON-BAITAH - “Methodology and Instruments of Building Automation and Information Technology for pervasive models of treatment and Aids for domestic Healthcare”. The authors thank Arturo Argentieri for technical support in the setup of the system presented in this work.

References