An Object- and Task-Oriented Architecture for Automated Video Surveillance in Distributed Sensor Networks

Eduardo Monari, Sascha Voth and Kristian Kroschel
Fraunhofer IITB - Institute for Information and Data Processing
Fraunhoferstr. 1, 76131 Karlsruhe, Germany
{eduardo.monari,sascha.voth,kristian.kroschel}@iitb.fraunhofer.de

Paper ID 49

Abstract

In this paper, an agent-based software architecture for automated wide area video surveillance systems is presented. The proposed concept is designed for detection and tracking of moving objects across multiple camera views. The surveillance system consists of a decentralized collaborative sensor network with object- and task-oriented architecture. At sensor node level, image processing algorithms are applied for event and object detection. In case of detection (e. g. motion) an agent-based multi-sensor processing cluster is created. Each instantiated cluster is responsible for observation of one object in the scene. Object handover is managed autonomously by the dynamic sensor clusters. The dynamic sensor clustering approach allows adding new sensors without resetting the system parameters, which is a big advantage in large sensor networks. Furthermore, by using the agent-based architecture it is possible to create a framework with an adaptive data and processing load. Additionally, upgrade of system capabilities can be done easily updating or adding new processing agents. The proposed concept has been proved on an experimental video surveillance system at the Fraunhofer IITB.

1. Introduction

Video surveillance and monitoring is one of the most recent fields of development and research. Due to the increasing threat by crime, industrial espionage and even terrorism, video surveillance systems became more and more important during the last years.

Most of the video surveillance systems, which are currently used, are managed by human operators who constantly monitor all video streams. This reduces the efficiency of the surveillance task (given by the vigilance of the operator) and limits the number of applicable video sensors.

One possible solution to overcome these restrictions is the use of automated surveillance and security systems. Such systems are able to operate 24h a day with constant performance and are able to manage a higher number of sensors at the same time.

Especially in large video sensor networks, there is an increasing need for high quality automated surveillance methods. When using a small number of sensors, the operator is able to overlook all videos simultaneously and to switch his focus between them to keep a person in view. A higher number of sensors require an abstract visualization of the scene for situation awareness (e.g. a map with all detected and tracked objects). Hereby the object detection and tracking task cannot be done by interaction of the operator. In an automated video surveillance system this task has to be fulfilled by integrated processing modules without the operator knowing about the participating sensors and needed data. In doing this the operator only receives processed information about events or location of objects and is then able to manage more complex situations. A major task for the automatic surveillance system is the multi-sensor multi-target tracking functionality. The tracking of objects with constant identity across different cameras views is a complex task.

In this paper we present an object-oriented concept for a distributed agent-based video surveillance system, able to detect and track multiple objects across different sensor views in a wide observation area. The focus hereby is the usability of the multi-sensor system on an IP-based surveillance network with a very high number of sensors. This paper is organized as follows: After a short overview of related works in chapter 2, chapter 3 provides a brief overview of the basic components of our system architecture. After

978-0-7695-3341-4/08 $25.00 © 2008 IEEE
DOI 10.1109/AVSS.2008.21

these chapters 4, 5 and 6 describe the basic components in detail. In a final step, the experimental system and future works are presented.

2. Related Work

Few approaches have been proposed for distributed agent-based surveillance systems [4], [16], [10], [1] and [5]. In [10] and [4] Collins et al. introduce a system for video surveillance and monitoring (VSAM) in large areas. The system consists of multiple calibrated cameras and a site model. The objects were tracked by using correlation and 3D location on the site model. The system architecture described in [10] and [4] is centralized, and therefore not fault tolerant and scalable with the number of sensor nodes [5].

In [16] Ukita et al. introduce a real-time cooperative multi-target tracking system. The system consists of a group of "Active Vision Agents" (AVAs), each of them connected to a dedicated active camera. All AVAs cooperatively track their targets by dynamically exchanging object information with each other. With this cooperative tracking capability, the system as a whole is able to track multiple moving objects.

However, the architecture described in [16], is able to track only one object (target) with each sensor. This is justified by the use of active sensors (pan/tilt). Tracking of numerous targets is solved by applying a high number of sensors for the area under observation, but this approach is typically not practicable for operational systems.

In [1] Atsushi et al. introduce a method for multi-target observation in wide area environments. The architecture proposed is highly flat and its components consist of distributed autonomous image processors (so-called "watching stations") and so-called "station parameter management agents". Each "watching station" is assigned to a dedicated sensor node but is able to apply multiple image processing algorithms ("seeing agents"). The "station parameter management agents" only provide sensor parameters and do not control the "watching stations". In this approach the main drawback is the required computational power for the distributed autonomous "watching agents". Each "watching agent" performs tracking algorithms, data fusion and decision making with high requests on computational power. Consequently, if multiple sensors observe the same object in the world, the system performs multiple complex processing tasks (one for each sensor).

3. Architecture Overview

The architecture proposed in this paper is led by the idea of an object- and task-oriented agent-based data processing. By "task-oriented" the coordination of available sensors and scheduling of computational resources depending on the surveillance tasks assigned by the security staff or human operator is meant. Hereby, the system does not observe all objects in view and consequently does not process all sensors simultaneously, but focuses on interesting events (look-out mode) and starts complex observation processes for detected relevant objects. In this case relevant sensors are assigned to clusters for multi-sensor processing and data fusion. The architecture consists of a hierarchical structure with two fixed and one dynamic level. The lowest level (fixed) summarizes the data sources (sensors) with associated *Specialized Detection Agencies* (SDAs). The SDAs perform low level object detection, segmentation and feature extraction. Each sensor is continuously processed independently by an integrated SDA. These agencies have observation functionality for further algorithms with higher requirements to the computational power (e. g. tracking or object recognition/identification).

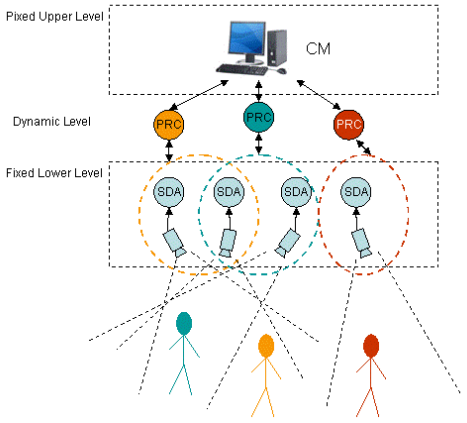

Figure 1. Overview of the object- and task-oriented system architecture with two fixed and one dynamic level. On the dynamic level, object-based *Processing Clusters* are applied for distributed collaborative multi-sensor processing.

The highest level node (fixed) is the so-called *Cluster Manager* (CM). The CM is responsible for the management of the medium dynamic level which consists of temporarily existing multi-sensor processes, so-called *Processing Clusters* (PRCs). The PRCs in turn are advanced processing modules which are identified with a specific surveillance task (e. g. tracking). They have the ability to manage and process multiple sensors (sensor clusters). Additionally, they are able to determine and manage autonomously those SDAs, which are needed to fulfil the associated surveillance task.

The collaboration of the components on the three levels can be described by the following example: If a new surveillance task has to be fulfilled, the human operator assigns the new task to the system, setting up a semantic description using a Human-Machine-Interface (HMI, e.g. application

frontend) interactively. The task parameters are interpreted by the *Cluster Manager* (CM) which now starts a task specific process (PRC) for advanced data processing.

In the next step, the new PRC uses the initial parameters provided by the human operator (e. g. start position of an interesting object) to determine the suitable sensors (SDAs) for the dedicated task. Additionally, it sets up connections to the involved SDA for further data requests.

Now, the subscribed SDAs continuously transmit all information about observations to the PRC for multi-sensor fusion and advanced processing (e.g. tracking). Over time, if due to object movement or changes of observation conditions other sensors or SDAs become more suitable or subscribed sensors are no more applicable for the running task, the PRC autonomously reorganize the sensor participants by subscribe to new SDAs and unsubscribe to other one.

The main advantages of this framework are network scalability, failure robustness of individual components and the possibility to reconfigure the system on- and offline (e. g. add/remove sensors). If we consider SDAs as parts of the sensors (e. g. intelligent cameras), the scalability of the sensor network will be mainly affected by the required computational power for the PRCs. This means, that the needed processor capacity is not affected by the number of sensors but by the number of average detections or events and the number of surveillance tasks.

Robustness is typically given by the redundant sensor clusters. If a sensor stops working, only a small field of view will be out of range. This lacking field of view is handled like an object occlusion by the PRCs. Finally, online reconfiguration is quite an important attribute for large surveillance systems. With the unfixed sensor assignment to local processing units and the dynamical cluster generation the participating sensors are redetermined periodically during the surveillance tasks of the PRCs. In doing this, after adding new sensors to the network and online registration of the sensor calibration parameters in a database, the new sensor is available to all existing PRCs for later cluster reorganisation.

In the following sections each type of level nodes (CM, PRCs and SDAs) is explained more detailed.

4. The Cluster Manager Architecture

The *Cluster Manager* is the highest level of the system architecture. It fulfils three main tasks. The first is the administration of the dynamic level which consists of the created PRC instances (Figure 2: PRC-Interfaces). Considering that there is one PRC for each relevant object in the scene the first task is equivalent to the administration of the objects or targets temporarily under observation.

As mentioned before, the system architecture introduced in this paper is led by the idea of an object- and task-oriented resource management for scheduling computational resources depending on the surveillance tasks. For the realization of this central idea, the *Cluster Manager* receives the information about observation jobs by the user over a Human-Machine-Interface (HMI, e.g. application frontend). In doing so, the *Task Management Interface* acts like an interpreter and enables the *Cluster Manager* to generate new PRC instances in case of new tasks formulated by the user.

A second functionality within the CM is the *Task- & Object-Control* module which buffers the assigned task parameters and takes over the task supervision for running jobs. This functionality includes the permanent observation of the running observation processes and the transmission of temporary observation/tracking results to external modules (e. g. situation awareness tools or visualization components).

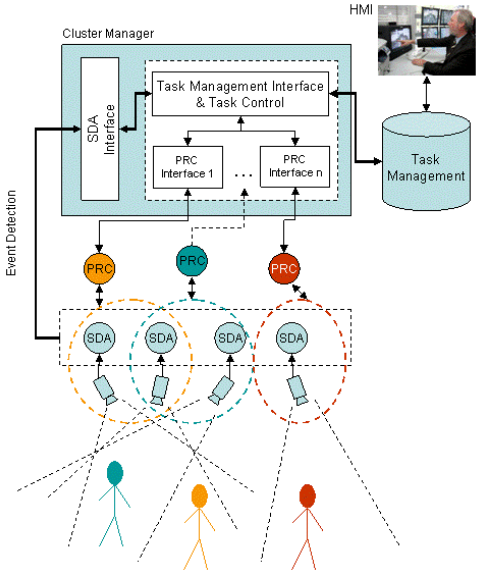

Figure 2. Internal Architecture of the Cluster Manager. The Task Management Interface & Task Control is responsable for the interpretation of tasks formulated by the human operator and for PRC scheduling.

As a last (optional) capability, the CM is able to process additional requests for new surveillance tasks coming from the SDAs. This option is interesting for automatic generation of specific PRCs triggered by detections from specific SDAs.

5. The Processing Cluster Architecture

Processing Clusters are the most complex modules with the highest requirements to the computational power. PRCs are responsible for the multi-sensor processing and fusion (e. g. tracking) of a dedicated object in the scene. They

include a *Dynamic Sensor Management* and the *Tracking Module* as their main components. The *Dynamic Sensor Management* (DSM) module is an independent component which determines cluster members for a given observation state. This determination is needed in case that the object under observation temporarily moves out of the field of view of the sensors in the cluster. Given the actual object parameters (position, motion model, object features etc.) and the a-priori known calibration parameters of all sensors, the DSM calculates the relevant subset of the sensor network that is needed for further observation of the object. This periodical dynamic reorganisation of the participating sensor nodes and the representation of the observation results in a common feature space lead to a decoupling of the sensor-related measurements and the multi-sensor data association process (e. g. tracking). So, if the sensors that collaborate in an observation cluster are substituted over time, only the appropriate measurements will change. The data association process for position fusion, however, is completely independent from the participating measuring sensors (Figure 4).

As mentioned before each PRC observes only one object in the scene (Figure 2). However, an object can appear in more than one sensor view at the same time. Therefore, the Tracking Module creates an active SDA-Interface for each cluster sensor (Figure 3). An additional sub module is responsible for fusion of observation measurements that are provided by the SDAs participating in the cluster. For a multi-sensor tracking application the fusion result is a unified trajectory and feature set for the observed object.

In doing so, a multi-target-tracking approach for each sensor is not needed in our system. The Tracking Module only consists of a single-target-tracker enables to track one object in a multi-target environment.

This is a main point where our system significantly differs from the architecture proposed in [1], which assumes tracking and multi-sensor fusion capabilities of each intelligent sensor node. The advantage of a sensor-oriented feature extraction and an object-oriented tracking improve exploitation of the computational resources.

5.1. Dynamic Sensor Management/Clustering

The object-oriented sensor clustering concept that was introduced before overcomes the typical object handover problem between sensor-oriented trackers. The PRCs involve only sensor nodes which could be relevant for a robust tracking task.

The sensor selection is done reiteratively by the *Dynamic Sensor Management* of the *Processing Cluster* (Figure 3). As in general a PRC is generated interactively by selection of initial parameters from sensor data (e. g. initial position of an object by selection of the object within a live video stream) at least one sensor is already assigned to the

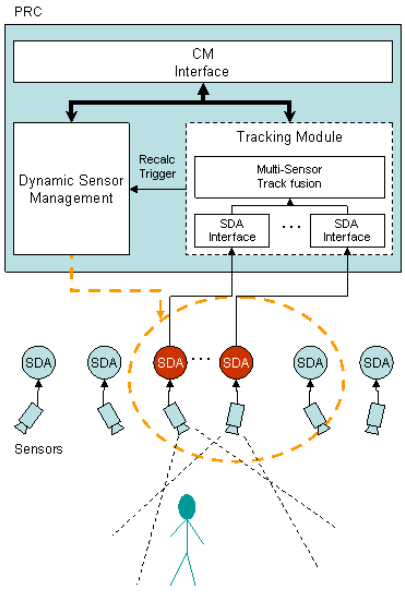

Figure 3. Architecture of the Processing Cluster (PRC). The two main modules are the *Dynamic Sensor Management* for continuous reorganization of the sensor-cluster memebers, and the *Tracking Module* for multi-sensor data fusion and tracking.

PRC after start. After this initialization step a cyclic object dependent reorganisation of the cluster participants is performed.

There are several algorithms in literature for camera-selection and clustering in sensor networks. Some approaches estimate the observation quality (e. g. quality of detected faces, person velocity in relation to the camera view, [15], [6]), and selects the cameras with the best quality coefficients. In [12] a look-up table is used for camera selection in large networks. Depending on the object position, the sensors a-priori assigned to this specific location are selected. In [14] the authors introduce a sensor selection method based on a new quality measure called "Appearance Ratio". The "Appearance Ratio" characterizes the object detection or segmentation quality of each sensor. Ercan et al. introduce in [3] a sensor selection method based on the minimum MSE of the best linear estimate of the object position as quality metric. The best linear estimate is defined by a chosen camera measurement model.

These methods are possible candidates as components of the Dynamic Sensor Management module. At the present state of development, a simple "k-nearest neighbour" approach is implemented which selects the k sensors with the field of views next to the object position. Unfortunately, this approach is only applicable if the sensor coverage is very high. In future works primarily modifications of the approaches presented in [14] and [3] will be evaluated in

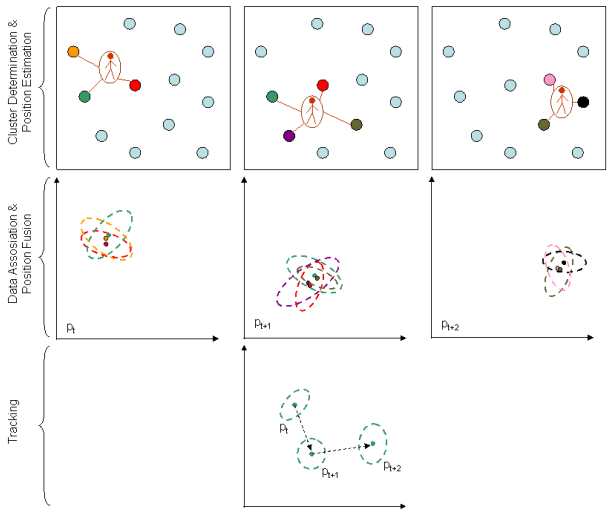

Figure 4. An exemplary dynamic clustering and data association process: The three pictures on top show the ideo of continuous cluster reorganization. The next row shows the observations of the current cluster members (assigned in a common feature space or coordination system). Finally, in each time step a data fusion approach is performed for object tracking.

our experimental video surveillance system. Enhancements are mainly needed, because of the non-overlapping field of views of our experimental sensor network.

Especially, we will analyse the use of sensor clusters for logical relationship between sensor nodes with disjoint field of views. As an example for the application of such clusters, cameras installed in front of elevator gates in a building, have to belong to an own cluster for continuous object tracking across different floors. This relationship can not be modelled with 2D representation of field of views.

5.2. SDA-Interfaces/Tracking- Module

Depending on the determined sensors in the cluster, instances of SDA-Interfaces are created for each cluster member. The SDA-Interfaces are active processes which subscibe to cluster SDAs, which now in turn start providing most recent data about observed objects (e. g. position, colour features, shape, position, etc.). Each time new observation data is available, a data validation and association process is performed by the Tracking Module for position estimation and identification of the observed object. In our PRCs we implement a data association approach based on Bayesian fusion with inconsistency detection, similar to the approach presented in [11]. All object positions provided by the observing SDAs are modelled as bivariate Gaussian distributions. Hereby, all SDAs provide the positions of all objects in the field of view. Using the approach in [11] the observations are evaluated and the inconsistency relative to the estimated position by the tracking approach is determined. Sensors which provide more consistent data are preferred during the Bayesian fusion step. The fused position measurement of an object is then used as the new measurement for a standard Kalman filter as linear motion estimator.

Further details about the position fusion methods will not be discussed in detail in this work, as in this paper we focus on the network architecture only. Nevertheless, we will point out, that there are numerous approaches for this task, discussed in many publications [2], [7], [13], [4], [9], [8].

6. Specialized Detection Agency

The *Specialized Detection Agencies* are distributed software applications with simple event notification functionality (trigger) and feature extraction capability. Each sensor in the network is directly assigned to a different SDA. The agencies in turn consist of three logical units. First of all, the *Sensor Interface* for raw data access (e. g. video streams); Second, the *PRC Interface* for communication to the subscribed PRCs (subscriptions); and last, the so-called *Simple Agent Plattform* (SAP). The SAP is a modular platform for dynamic generation of vision agents with different capabilities. Depending on the available processor resources on the intelligent sensors or the specific detection task ordered by a PRC, the SAP activates one or multiple vision agents for image processing. All activated vision agents perform individual algorithms for event detection and feature extraction (e. g. motion detection, face detection, change detection (abandoned luggage, theft)). After this the extracted information and object features are available to all subscribed PRCs. By providing observation descriptions on feature level, and additionally only on demand the network load is reduced significantly. This guarantees that only task-relevant processes affects network load by extensive data exchange.

The object description messages are structured in a header and a body part. The message header includes object independent sensor and observation parameters like SensorID, calibration parameters, timestamp, activated detection agent (e. g. motion detection), body size etc. The body on the other hand, includes feature vectors about the observed objects. At the present state, our experimental system includes motion detection agents for the object tracking task only. Hereby, the body includes the positions of the objects as bivariate Gaussians (in a common coordinate system), height, width and the estimated object type (human/man-made object), for all detected objects in sensor range as default features. Additionally, an optional data field for object descriptors like color histograms, contours, face templates, covariance descriptors etc. is provided by the protocol for advanced data association and fusion approaches.

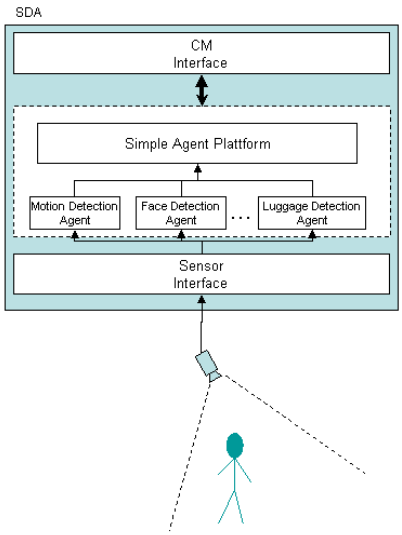

Figure 5. Specialized Detection Agency. The SDA is a sensor related process, therefore only one sensor interface is included. With the *Simple Agent Platform* the SDA is able to perform one or more detection approaches on the sensor data and provide the observations to subscribed PRCs over the *PRC-Interface*.

Figure 6. SDAs of two cameras performing motion detection and position estimation. The green lines visualize the estimated height and width of the detected blob. The red cross shows the estimated position of the object blob, after shadow removal.

The decentralized detection agencies enable to distribute computational load and enable the system to efficiently transmit data from heterogeneous sensors and approaches. Furthermore, the distributed structure allows to highly reduce the data load on the network and at the same time to focus computational load on external processors.

7. Experimental System

The experimental system at the Fraunhofer IITB consists of about 25 cameras installed on three floors in our building. All sensors are state-of-the-art IP-cameras, and commercially available. The fields of view of the sensors cover a major part of the test areas, but not completely. Therefore, the vision system needs robust algorithms that are able to track objects in case of occlusions or disappearance and reappearance. This capability is mainly needed to track objects or people across different floors (using elevators).

For evaluations and tests of the system capabilities we installed four cameras in the entrance hall, one of those in particular for face detection of people entering the building. Two more cameras observe the lounge at the second floor in front of a conference room. About 12 more cameras cover the corridors of several floors and one office wing of our institute. Furthermore, we installed additional video cameras observing the elevators on all three floors. Object classification and face recognition approaches will be applied on these cameras for tracking persons who enter the elevator at one floor and leave it at another one.

The computational capacity of the experimental systems consists of 9 high-end PCs available on the market. Five of them are temporarily emulates the intelligent cameras (with 5 running SDAs each). One processor core is actually used for the CM. The other processor cores provide computational resources for PRCs which means for surveillance tasks.

We test the architecture for single-object and multi-object tracking, which means, using one or more PRCs at the same time and observed a quite stable performance with low network load. All 25 SDAs have processed the assigned video stream with 10fps and 4CIF resolution and therefore generated object observation messages with about 10KB per frame and object in range. Consequently, the amount of data received by a PRC with an average sensor cluster size of 3 is approx. 300KB/s which is negligible. On the other hand, we observe, that the CPU load needed by a PRC for data reception, synchronization in parallel, data association, fusion and tracking is very high.

For proof of concept, figure 6 and 7 show the results of four SDAs performing motion detection and position esti-

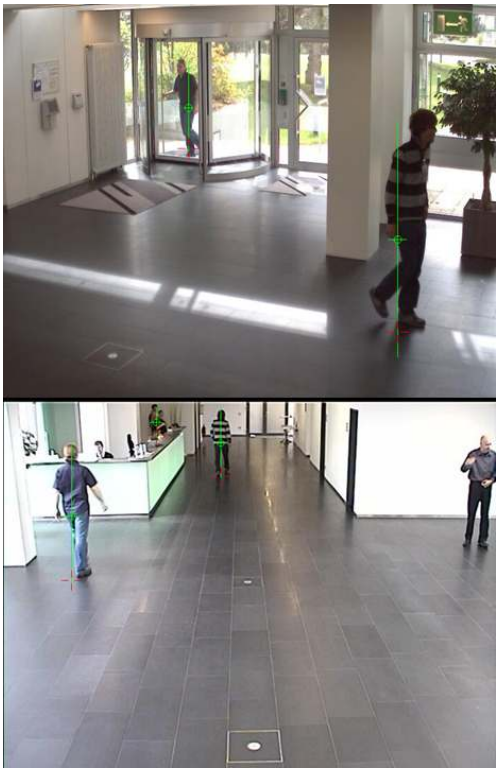

Figure 7. SDA of two more cameras at the first floor of the IITB.

mation to an object of interest. In this sequence one target has been tracked on a corridor at our lab, over seven cameras with overlapping field of view. In addition to the motion detection capabilities, the SDAs include a calibration parameter sets for the dedicated cameras, which enable the SDAs to transform the geometric feature extracted from moving blobs (height, width, position) in a global coordination system. The estimated features are illustrated by the green lines to each object.

Figure 8 shows the observed detections of the objects in a common coordination system. All detections are visualized as marker with different colors for different SDAs. As shown in the plot, the detections of each sensor are limited to certain areas (given by the FOV of the cameras). The trajectory of the observed object generated by the assigned PRC is illustrated by the green solid line.

Even though our main focus for application is object tracking, the system architecture presented in this paper is not application specific. For example it would be also thinkable to perform a ”person identification”-task to search a specific person in large camera networks. For this task, the human operator might provide one or more face template(s) to a ”face recognition”-PRC as initial parameters. The SDAs in this case perform face detection and feature extraction.

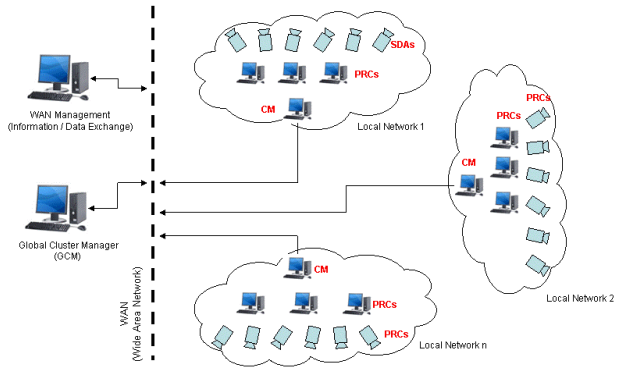

Figure 9. Expandability of the System. *Cluster Managers* are responsable for task handling with a dedicated number of SDAs. Migration of single tasks (PRCs) between local networks is managed by the *Global Cluster Manager*.

8. Future Work

Especially for large area surveillance systems expandability is an important aspect. The architecture presented in this paper is highly expandable of two reasons: First, the SDAs as distributed processing units, do not directly charge the computational resources of the system. Consequently, adding sensors by constant number of observation tasks does not significantly increase the system load. Second, the PRCs are only generated for dedicated surveillance tasks (assigned by the human operator). In doing so, the computational load of the system mainly depends on the amount of observation tasks. Nevertheless, a higher number of sensor nodes (SDAs) implies a higher load for the PRCs. As mentioned before, the PRCs dynamically perform a reorganisation of the cluster which leads to a higher complexity for a higher density of sensors.

For very large systems with local operators responsible for different areas, the system is designed to be subdivided in ”local networks”, with a dedicated number of SDAs and a CM each. An additional *Global Cluster Manager* (GCM) is integrated in the architecture for communication between the local CMs and the capability for migration of single PRCs from a subnet to another (for tracking object crossing different subnets).

For simulation of such subnets, the experimental system at our institute will be logically divided in three local networks - one at each floor - with at least six SDAs each. The WAN is simulated by the network backbone at our institute. Because of the identical hardware setup and infrastructure, we will be able to determine a realistic performance gain compared to the basic architecture.

A second ambition in future works will be the integration of heterogeneous sensors (and SDAs) in the experimental system, to prove the sensor independent concept. We intend

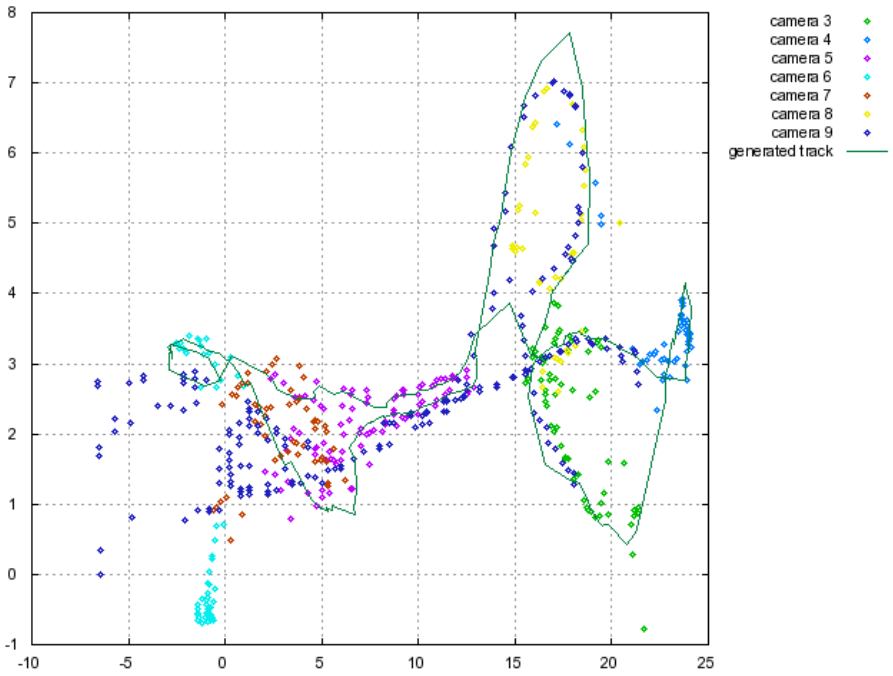

Figure 8. Sensor observations in a multi-target environment (dots), and the estimated track for an assigned moving target (green line).

to integrate microphones for acoustic localization of objects with proper SDAs.

References

[1] N. Atsushi, K. Hirokazu, H. Shinsaku, and I. Seiji. Tracking multiple people using distributed vision systems. *Proceedings of the IEEE Int. Conf. on Robotics & Automation*, 2002. 2, 4

[2] Q. Cai and J. K. Aggarwal. Tracking human motion in structured environments using a distributed camera system. *IEEE Tans. on PAMI*, 2(11), 1999. 5

[3] A. O. Ercan, A. E. Gamal, and L. J. Guibas. Optimal placement and selection of camera network nodes for target localization. *Proceedings of the IEEE Conf. on Advanced Signal Based Surveillance*, 2003. 4

[4] H. Fujiyoshi, R. Collins, A. Lipton, and T. Kanade. Algorithms for cooperative multisensor surveillance. *Proceedings of the IEEE*, 89(10), 2001. 2, 5

[5] A. Goradia, N. Xi, M. Prokos, Z. Cen, and M. Mutka. Cooperative multi-target surveillance using a mutual analysis approach. *The IEEE/ASME Int. Conf. on Advanced Intelligent Mechatronics*, 2005. 2

[6] K. Huang and M. Trivedi. Networked omnivision arrays of intelligent environment. *Proceedings of the Applications and Sciences of Soft Computing IV*, 2001. 4

[7] T. Huang and S. Russell. Object identification in a bayesian context. *Proceedings of the IJCAI*, 1997. 5

[8] O. Javed, Z. Rasheed, O. Alatas, and M. Shah. Knightm: A real time surveillance system for multiple overlapping and non-overlapping cameras. *The fourth Int. Conf. on Multimedia and Expo (ICME)*, 2003. 5

[9] O. Javed, Z. Rasheed, K. Shafique, and M. Shah. Tracking across multiple cameras with disjoint views. *Proceedings of the 9. IEEE Int. Conf. on Computer Vision*, 2003. 5

[10] T. Kanade, R. T. Collins, and A. J. Lipton. Advances in cooperative multi-sensor video surveillance. *Proceedings of DARPA Image Understanding Workshop*, 1:3–24, 1998. 2

[11] M. Kumar, D. P. Garg, and R. A. Zachery. A generalized approach for inconsistency detection in data fusion from multiple sensors. *Proceedings of the American Control Conference*, 2006. 5

[12] J. Park, C. Bhat, and A. C. Kak. A look-up table based approach for solving the camera selection problem in large camera networks. *Workshop on Distributed Smart Cameras (ACM SenSys'06)*, 2006. 4

[13] R. Romano, L. Lee, and G. Stein. Monitoring activities from multiple video streams: Establishing a common coordinate frame. *IEEE Trans. on PAMI*, 2(8):758–768, 2000. 5

[14] L. Snidaro, R. Niu, P. K. Varshney, and G. L. Foresti. Automatic camera selection and fusion for outdoor surveillance under changing weather conditions. *Proceedings of IEEE Int. Conf. on Distributed Computing in Sensor Systems*, 2003. 4

[15] K. Triverdi, M. Huang, and I. Mikic. Intelligent environments and active camera networks. *IEEE Transactions on Systems, Man and Cybernetics*, 2000. 4

[16] N. Ukita. Real-time cooperative multi-target tracking by communicating active vision agents. *Ph. D. Thesis, Kyoto University, Japan*, 2001. 2