

# Task-Oriented Situation Recognition

Alexander Bauer\*<sup>a</sup>, Yvonne Fischer <sup>a,b</sup>

<sup>a</sup>Fraunhofer IOSB, Fraunhoferstrasse 1, 76131 Karlsruhe, Germany

<sup>b</sup> Karlsruhe Institute of Technology (KIT), Germany

## ABSTRACT

From the advances in computer vision methods for the detection, tracking and recognition of objects in video streams, new opportunities for video surveillance arise: In the future, automated video surveillance systems will be able to detect critical situations early enough to enable an operator to take preventive actions, instead of using video material merely for forensic investigations. However, problems such as limited computational resources, privacy regulations and a constant change in potential threads have to be addressed by a practical automated video surveillance system. In this paper, we show how these problems can be addressed using a task-oriented approach. The system architecture of the task-oriented video surveillance system NEST and an algorithm for the detection of abnormal behavior as part of the system are presented and illustrated for the surveillance of guests inside a video-monitored building.

**Keywords:** Video surveillance, situation recognition, abnormal behavior detection, syntactic pattern recognition

## 1. INTRODUCTION

During the last years, video surveillance systems became more and more important. Recently, there are a lot of new developments focused on automated video surveillance which are often highly adapted to a specific application and strongly coupled to characteristics of the used sensors and its signal processing algorithms. However, for a general approach to situation recognition, the type of sensors used in the surveillance system should not be relevant, as situations are best described in terms of objects, their attributes and interactions over time. For this reason, in this paper we propose a task-oriented architecture for situation recognition which separates the signal-processing level from the semantic level. The connection between these levels is realized through an abstract description of the task-relevant objects in the supervised area, the object-oriented world model. As an application of the task-oriented situation recognition approach, an implementation for the detection of abnormal behavior of a person's movement in a building is presented.

## 2. RELATED WORK

Several new developments in automated video surveillance systems can be found in the literature<sup>1, 2</sup>. For example, Monari et al.<sup>1</sup> present an object-oriented concept for detecting and tracking moving objects across multiple cameras. Based on these object assessments, there is a need for automatic situation recognition or event detection for context-based interpretation<sup>2</sup>. Situation analysis can be associated with higher-levels of data fusion like level 2 and 3 of the JDL data fusion model<sup>3</sup>. Basic concepts and approaches to these higher levels are also discussed by Jakobson et al.<sup>4</sup>. In a video surveillance system, the main focus in modeling situations lies on the detection of abnormal behavior for supporting situation awareness. The advantage of some existing behavior models like Xiang et al.<sup>5</sup> is the unsupervised learning technique, but often they are based on scene-events that are close to signal level and are therefore dependent on the sensor characteristics.

\*alexander.bauer@iosb.fraunhofer.de; phone +49 (0)721 6091-395; fax +49 (0)721 6091-233; www.iosb.fraunhofer.de

### 3. TASK-ORIENTED INFORMATION PROCESSING

In a task-oriented approach to information processing, only task relevant information is acquired, analyzed, stored and processed. To provide an extendable framework for new types of tasks, a service oriented architecture (SOA) has been chosen for the implementation of the task-oriented surveillance system NEST (Network Enabled Surveillance and Tracking)<sup>6</sup>. For each task type, a set of modular services for situation recognition is assembled. If a task template is instantiated by an operator, the services are activated and start to perform the task autonomously, until a task-relevant event occurs which has to be handled by the operator.

To be able to efficiently define new services which encapsulate the detection and recognition of task-relevant situations, the signal level processing has to be modularized into *low-level services*, which provide capabilities to acquire information from sensor systems and produce object level information. This approach greatly reduces the effort of developing services for new situations (*high-level services*), as the modeling of situations can be done in terms of the spatio-temporal configuration of objects and their features.

To bridge the gap between low-level services working on the signal level and high-level services for situation analysis on the object level, an *object-oriented world model* (OOWM)<sup>7</sup> has been developed. It is responsible for the fusion of object observations from low-level services into a consistent interpretation of the presence and features of task-relevant objects. High-level services communicate their information requirement to the OOWM. A *low-level service planner* triggers low-level services to satisfy the requirements, based on the representation of uncertainty about task-relevant object features in the OOWM. Figure 1 sketches the overall system architecture.

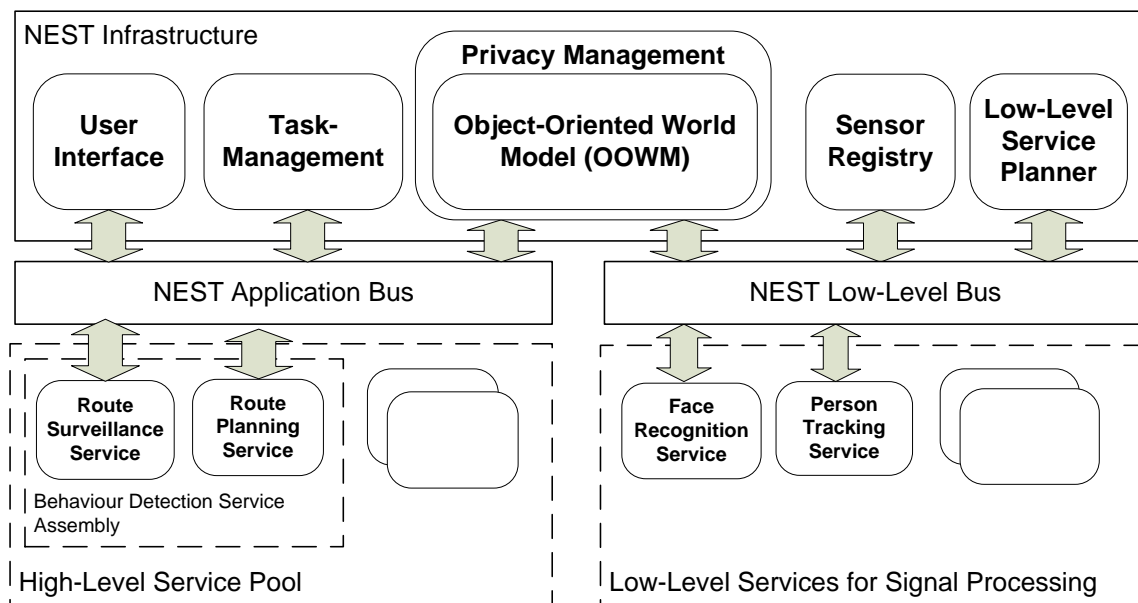


Figure 1: Task-Oriented System Architecture

The task-oriented approach provides several advantages, of which three are highlighted here:

- **Expandability** – New requirements on the surveillance system are easily covered by defining additional surveillance tasks. Existing services for high-level and low-level processing can be reused. As new sensor systems are implemented as services too, by adding a new sensor service, all existing high-level services profit from the additional confidence introduced into the system, as the signal level processing is decoupled from the high-level interpretation of object level information.
- **Efficiency** – As only task-relevant information needs to be gathered, computational resources for signal analysis only need to be occupied if they are needed to acquire information on task-relevant objects.

- Privacy Protection – Sensor signals (e.g. video streams) are only processed, if a task-relevant information requirement is present. No processing or storage is necessary if not a security relevant task exists to justify it.

As an example for a high-level service for abnormal behavior detection, the route surveillance service is explained in detail.

#### 4. ABNORMAL BEHAVIOUR DETECTION

In general, abnormal behavior is defined as a deviation from statistical norms. For modeling human behavior or human activities, three different approaches can be found in literature<sup>8,9</sup>:

- Statistical approach: Traditional methods for this kind of approach are graphical models like Bayesian belief networks or dynamic Bayesian belief networks, which also make use of temporal dependencies. The simplest form of a dynamic Bayesian network is the hidden Markov model. For each behavior or activity, one statistical model is generated, which is usually done with learning techniques. As it is not always possible to generate a model of the abnormal behavior, only normal behavior can be modeled and any behavior that deviates from the normal model can be classified as abnormal behavior.
- Syntactic approach: Syntactic approaches define the abnormal behavior patterns explicitly. This approach is straightforward and it is not necessary to generate a lot of learning data. However, it must be possible to accurately describe the behavior patterns that should be detected as abnormal. The behavior pattern can for example be described by the use of grammars and the detection of a special pattern is done by grammar parsing.
- Knowledge- and logic-based approach: These approaches make use of the domain knowledge, which can be modeled with ontologies. Ontologies are standardized representations of the human activities and are therefore independent of the algorithmic choices. Abnormal behavior can be detected with logical reasoning techniques.

In this article, we used the syntactic approach with the use of grammars, since the abnormal behavior we want to detect are directly describable. We only make use of a person's position to infer several discrete state estimates, which can be easily modeled by a sequence of symbols. Therefore, abnormal behavior can be detected by grammar parsing methods.

#### 5. ROUTE SURVEILLANCE SERVICE

To illustrate the implementation of a situation recognition service in a task-oriented surveillance system, a service for the detection of abnormal behavior in an indoor scenario is explained in detail. In this scenario only a single person (e. g. a guest at our institute) per task has to be supervised during his walk from the reception desk at the entrance to its designated destination (e. g. a conference room or office of an employee). If several persons have to be supervised, multiple independent tasks are started. The security officer enters the information about the guest (reason for the visit, destination, etc.) and starts the surveillance task by selecting the person in an initial video stream. After the guest leaves the reception area, the NEST systems takes over the surveillance task and starts/activates all necessary services for job processing in the background. In particular, only sensors needed to track the person are activated and processed, while at the same time, only data related to observed persons are stored temporarily. That is, video data of other people walking in the same protected area is not stored, because no surveillance task exists for these objects. As long as the guest is walking on the dynamically allowed paths, no alert is released by the NEST system. When the guest arrives at the designated destination, the NEST system sends a note to the security officer and the services related to the surveillance task are terminated. In the case that the observed person enters forbidden areas (e. g. wrong floors, classified areas/rooms, etc.) an alert is emitted to the NEST user interface.

##### 5.1 Architecture

To implement this task, a high-level service has been developed. The *Route Surveillance Service (RSS)* automatically evaluates constraints on the person's movements, such as possible paths to the destination, restricted areas, expected time to reach the destination and the certainty about the person's location. If one of the constraints is violated, the service

generates an alarm of abnormal behavior of the supervised person. For each person to be supervised, a new RSS instance is created and configured with the task-specific parameters (person's, destination, allowed areas and system internal ID).

In order to formally describe abnormal behavior, a structural approach for pattern recognition is chosen. In contrast to learning-based methods, it allows modeling of situations in a human understandable manner: A set of grammars of discrete states of the supervised person during its presence in the building defines abnormal behavior. The time-series of states is matched against the grammar in order to generate alarms. The state of the person is described as the discrete state vector of the following parameters:

- *Current area state*: allowed (A), forbidden (F), unknown (U)
- *Current transition state*: path forward (F), path reverse (R), neutral (0)
- *Movement state*: standing (S), walking (W), running (R)
- *Observation state*: observed location (O), inferred location (I), unknown location (U)

Figure 2 illustrates the processing chain of the RSS. The process involves three subsequent processing steps, each performed by different modules: a filtering and spatial inference module (FSIM), a discrete state estimation module (DSEM) and a state-time-series analysis module (STAM). The processing chain is executed at a frequency of 1 Hz.

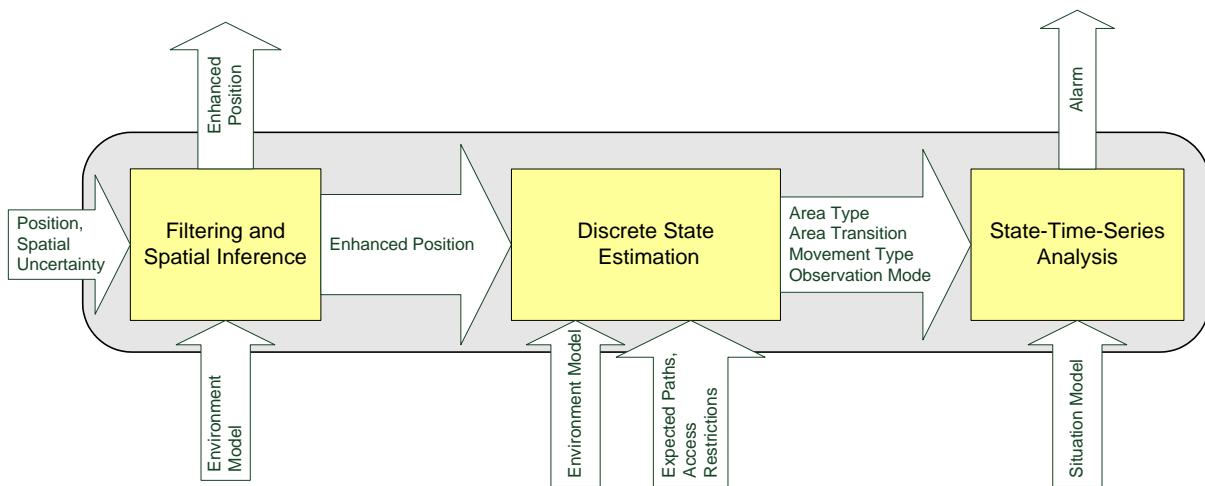


Figure 2: Processing Workflow of the Route Surveillance Service

## 5.2 Filtering and Spatial Inference Module (FSIM)

The FSIM is responsible for the estimation of the persons' position and movement. Task-independent estimation of the persons' position is performed at the OOWM using state-of-the-art data association and fusion methods<sup>10</sup> based on observations delivered by low-level services. However, these methods are only able to estimate the position of a person in areas where sensors are available to monitor it. For the task-specific detection of intrusion into restricted areas, it is important to detect as well if the person has entered a restricted room bordering a monitored room. Therefore, the RSS exploits an environmental model of the building to infer the position using spatial inference. The environmental model is available to all services of the NEST system using a server implementing the web feature service (WFS)<sup>11</sup> interface definition. Rooms and connecting doors are represented as geospatial features inside the WFS server and spatial queries can be executed to retrieve relevant features.

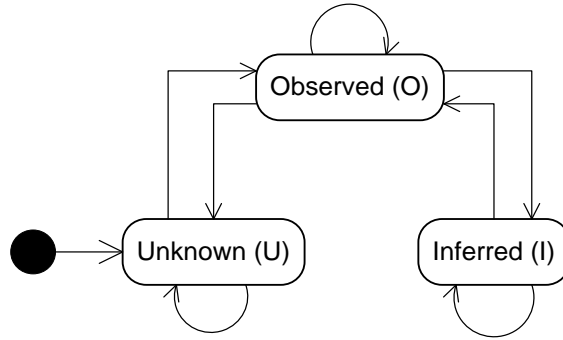


Figure 3: States and transitions of the FSIM module, representing the current knowledge about the person's position.

To receive track information from the OOWM, the RSS subscribes to a periodic update of the person's dynamic information using a web-service interface of the OOWM. Track information is delivered at a frequency of 1 Hz in terms of the person's estimated position, velocity and an error covariance matrix representing the multivariate Gaussian error distribution. If no sensor is currently observing the person, the error matrix coefficients grow quickly according to the underlying movement model. The knowledge about the current position is represented by different states of the FSIM and their transition can be described by a state machine (see Figure 3). In the *Observed* state, short periods of missing visibility of the person are tolerated, but as soon as the error matrix coefficients reach a predefined threshold, the FSIM tries to infer the position based on the environmental model. If the position can be inferred from the environmental model (*Inferred* state), the inferred position instead of the observed position is delivered to the DSEM. Otherwise, the position is estimated to be unknown (*Unknown* state). As soon as new observations become available and the error matrix coefficients decrease below threshold, the FSIM will switch back to *Observed* state. The state of the FSIM determines the *Observation state* vector and is later on used for time-series analysis.

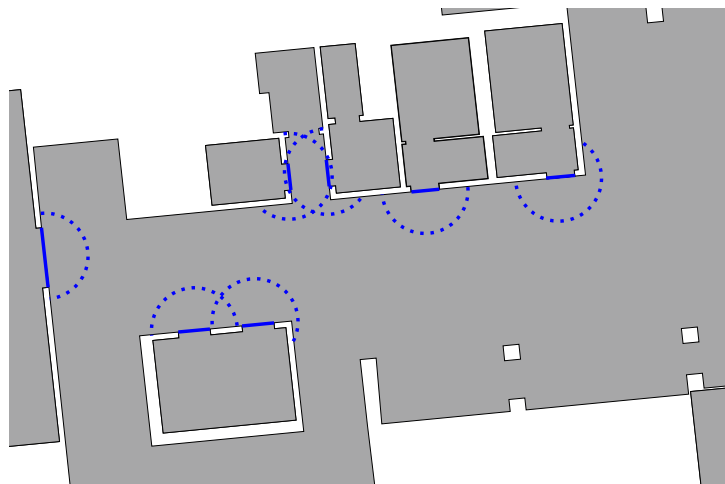


Figure 4: Catch-regions for the detection of entrance into unmonitored rooms.

In order to detect entrance into an unmonitored room, catch-regions are defined around doors of the current room, as illustrated in Figure 4. Anytime a person's position becomes unknown inside a catch-region, it is inferred that the person entered an unmonitored room behind the door and the person's position is estimated to be approximately in the centroid of the unmonitored room.

### 5.3 Discrete State Estimation Module (DSEM)

Based on the position estimated from the FSMI module, the DSEM classifies the position and movement of the person. Allowed paths and corresponding allowed rooms leading to the person's destination are calculated by a *route planning service (RPS)* and handed to the RSS as parameters for each task instantiation. Paths and areas are encoded as LINESTRING and MULTIPOLYGON representations using the WKT encoding of geoshapes defined in the OGC WFS specification. The *current area state* is determined by matching the estimated position with allowed areas. The current room is determined based on the environment model. If the room has changed since the last execution of the DSEM, the direction of transition is matched with allowed path directions and results in the determination of the *current transition state* (forward, reverse or neutral). The movement of the person is classified according its velocity  $v$  ( $v < 0.2$  m/s: Standing,  $0.2 \text{ m/s} < v < 1.5$  m/s: Walking,  $v > 1.5$  m/s: Running). If the current position of the person is determined by the FSMI to be unknown, the current area state is asserted to be unknown and the current transition state is set to be neutral.

### 5.4 State-Time-Series Analysis Module (STAM)

The state vector defined in section 5.1 at each cycle is input for the STAM. All state vectors are stored as a state-time series. It represents the observed behavior of the person and the level of observation during the whole task execution at a high level of abstraction. In order to match the state-series against templates, the state-series is expanded as character string. The realizations of each state parameter are represented by single characters (see Section 5.1) and the full state vector is represented as the corresponding character concatenation. State vectors are delimited by a hyphen. An example for a normal behavior sequence is given as:

A0SO-A0SO-A0WO-A0WO-A0WO-A0WO-A0WO-A0WO-AFWO-A0WO-A0WO

The sequence represents 10 seconds of the time series, for a person commencing to walk towards his destination. In the case of intrusion into a restricted but unmonitored area, the sequence looks like:

A0WO-A0WO-F0WI-F0WI-F0WI-F0WI-F0WI-F0WI-F0WI-F0WI

Different behaviors result in different sequences. To represent classes of sub sequences which indicate suspicious behavior, a formal language approach is chosen. Regular expressions are the simplest representation of a formal language. They have been first studied by Kleene in the context of neural networks<sup>12</sup> and are still subject of research today<sup>13</sup>. They represent grammars of type-3 in the Chomsky hierarchy and it can be shown that they can be parsed by a finite state automaton<sup>14</sup>. The ability to represent a large variety of character sequences using short expressions makes them the tool of choice in software engineering for parsing of input strings and therefore implementations are readily available in any programming language. Table1 lists an exemplary set of typical suspicious sequences the RSS should detect and on which occurrence the operator should be notified. The notation is based on the regular expression syntax defined in the IEEE POSIX standard.



## 6. EVALUATION OF HIGH-LEVEL SERVICES

To evaluate high-level services for situation recognition, it is both necessary to have a realistic dataset of situation examples (scenarios) and to be aware of the influence of imperfect signal processing in low-level services. To control both parameters, a scenario simulator for the simulation of situations and sensor configurations has been developed. The simulator is able to simulate the dynamics of multiple objects, their attributes and resulting observations by sensor systems. For each object, static object attributes and the trajectory of movement in the building is defined. Sensors and corresponding signal processing methods are simulated based on the definition of footprint, extractable attributes and error distributions. During execution of the simulator, movements of the objects are simulated and imperfect sensor observations are generated. These observations are delivered to the OOWM for data association and fusion. For the evaluation of the RSS, the RSS is tasked to supervise a specific person modeled in the simulation and the resulting alarms generated are compared to the ground-truth of the modeled scenario.

Figure Y shows the display of the scenario simulator. The blue regions represent footprints of CCTV-cameras, as they are installed in the NEST demonstration system at Fraunhofer IOSB. It is easily seen that total coverage of all areas of the building can only be achieved by placing a lot of cameras with different view angles.

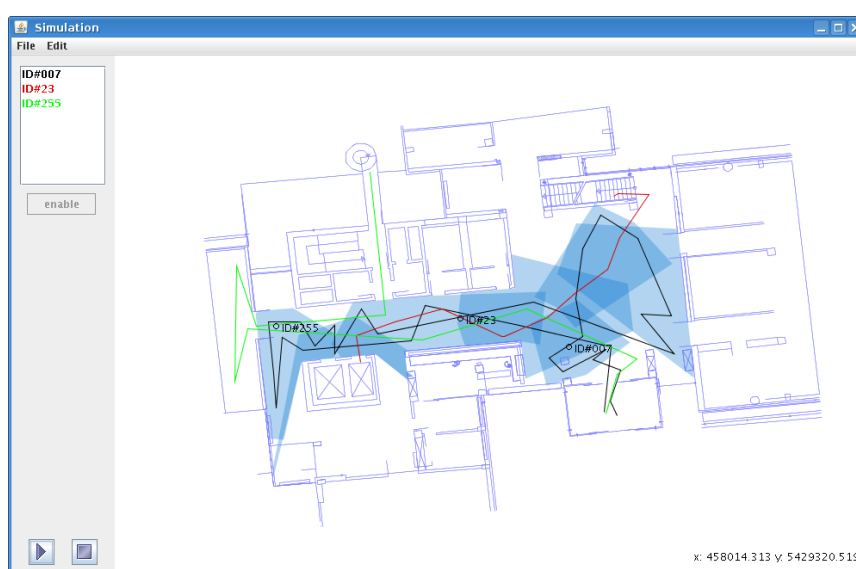


Figure 6: Scenario simulator for the simulation of object movements, sensor coverage and sensor processing.

A qualitative evaluation of the RSS using the simulator has shown that the presented RSS service could be one of the first services to be implemented as soon as robust person tracking in indoor scenarios are available. It is able to detect some of the most crucial events (disappearance of the person, tracker malfunction, intrusion into restricted areas etc.) which can be described by the development of the trajectory over time in terms of state-series. Of course, more complex situations involving multiple objects such as fighting, stealing and left-luggage detection require more sophisticated processing both on lower and higher levels of abstraction.

## 7. CONCLUSION

In this article a task-oriented approach to situation recognition is presented. It focuses the employment of sensor signal analysis and situation recognition algorithms on relevant surveillance tasks. Surveillance tasks are defined in a surveillance system based on a service-oriented architecture, as assemblies of high-level services. A Route Surveillance Service (RSS), a high-level service for the recognition of abnormal behavior of a guest inside a surveyed building is presented. A structural pattern matching approach using a grammar for abnormal behavior is employed to classify the

behavior of the guest. It is able to detect simple events such as intrusion into forbidden areas, tracker loss and person disappearance and similar events which can be described by the presented regular expression matching method.

## REFERENCES

- [1] Monari, E., Voth, S. and Kroschel, K., "An object- and task-oriented architecture for automated video surveillance in distributed sensor networks," IEEE International Conference on Advanced Video and Signal based Surveillance AVSS, 339-346 (2008).
- [2] Hampapur, A., Brown, L., Connell, J., Ekin, A., Haas, N., Lu, M., Merkl, H., Pankanti, S., Senior, A., Shu, C. and Tian, Y. L., "Smart video surveillance: exploring the concept of multiscale spatiotemporal tracking" IEEE Signal Processing Magazine 22(2), 38-51 (2005).
- [3] Steinberg, A. N., Bowman, C. L. and White, F. E., "Revisions to the JDL Data Fusion Model," Proc. SPIE 3719, 430-441 (2004).
- [4] Jakobson, G., Buford, J. and Lewis, L., "Situation Management: Basic Concepts and Approaches," [Information Fusion and Geographic Information Systems], DOI: 10.1007/978-3-540-37629-3\_2, Springer, 18-33 (2007)
- [5] Xiang, T. and Gong, S., "Video Behavior Profiling for Anomaly Detection," IEEE Transactions on Pattern Analysis and Machine Intelligence 30(5), 893-908 (2008)
- [6] Moßgraber, J., Reinert, F., Vagts, H., "An Architecture for a Task-Oriented Surveillance System," In Press.
- [7] Bauer, A., Emter, T., Vagts, H. and Beyerer, J., "Object oriented World Model for Surveillance Systems," Future Security: 4th Security Research Conference, , Fraunhofer Verlag, 339–345 (2009).
- [8] Turaga, P., Chellappa, R., Subrahmanian, V.S., Udrea, O., "Machine Recognition of Human Activities: A Survey", IEEE Transactions on Circuits and Systems for Video Technology, vol.18, no.11, 1473-1488 (2008).
- [9] Ryoo, M.S, Aggarwal, J.K, "Semantic Representation and Recognition of Continued and Recursive Human Activities", International Journal of Computer Vision, vol. 82(1), 1-24 (2009).
- [10] Mitchell, H. B., [Multi-Sensor Data Fusion: An Introduction], Springer, 173-200 (2007).
- [11] OGC, "Web Feature Service (WFS) Implementation Specification (Version 1.0.0)," OpenGIS®-Project Document OGC 02-058, OGC, (2001).
- [12] Kleene, S., "Representation of Events in Nerve Nets and Finite Automata," Princeton University Press,, 3-42, (2001)
- [13] Gruber, H., Holzer, M. "Finite Automata, Digraph Connectivity, and Regular Expression Size," Proceedings of the 35th international colloquium on Automata, Languages and Programming, Springer, Berlin, 39-50 (2008).
- [14] Bunke, H., [Syntactic and Structural Pattern Recognition Theory and Applications: Theory and Applications], World Scientific Pub Co Inc, (1988)