

# Diamond Sentry: Integrating Sensors and Cameras for Real-Time Monitoring of Indoor Spaces

Pavan Turaga, *Member, IEEE* Yuri Ivanov, *Senior Member, IEEE*

**Abstract**— Video-based surveillance and monitoring of indoor spaces such as offices, airports and convenience stores has attracted increasing interest in recent years. While video proves useful for inferring information pertaining to identities and activities, it results in large data overheads. On the other hand, motion sensors are much more data-efficient and far less expensive, but possess limited recognition capabilities. In this paper, we describe a system that integrates a large number of wireless motion sensors and a few strategically placed cameras and its application to real-time monitoring of indoor spaces. The system described here responds to an event immediately as it happens and provides visual evidence of the location of the event, thereby establishing an awareness of the events in the entire location being monitored, supplying the user with the information about ‘when’, ‘where’ and ‘what’ happens in the space as the events unfold. We introduce a system that is designed for maximizing the utility of the video data recorded from a location. It achieves this goal by following the minimal commitment strategy, where no data is discarded and no particular hypothesis is pursued until the time when the interpretation is necessary. Additionally, we employ an alternative modality to help in indexing video data for real time as well as for possible future use in forensic mode. We use the motion sensor data to specify policies of camera control. Utilizing these policies makes the application of machine learning and computer vision techniques simple to use to perform on-line surveillance tasks in a fast, accurate and scalable way.

**Index Terms**— Motion Sensors, Camera Network, Fusion, Surveillance, Activity Recognition, Human Detection.

## I. INTRODUCTION

Real-time surveillance and monitoring of indoor spaces has been gaining importance in recent years. Today one can see surveillance installations in a wide variety of settings – at homes, offices, airports, trade shows etc. Traditional systems rely on a single modality - most often video, occasionally augmented with audio. Such video-based systems generate massive amounts of visual data which brings challenges of storage, retrieval and event detection in large datasets to the forefront of science. Computer vision algorithms to detect events or persons are either not fast enough for use in a real-time system or do not have sufficient accuracy to singularly rely upon them. These challenges call for new modalities of sensing and data collection, which provide information about the monitored space that is rich enough to make intelligent judgments and draw inferences without drowning the system in a glut of data. To this end, we propose the use of a network of impoverished sensors spread throughout the indoor

space - specifically, we use wireless motion sensors. Such impoverished sensors provide low-bit rate information are significantly less expensive than cameras. But, they are ‘blind’ to important tasks such as human detection and recognition. Hence, we propose to integrate a large number of motion sensors with a few strategically placed cameras to monitor indoor spaces. Such a system offers significant new capabilities that would otherwise be impossible to obtain from only one of the modalities. Each modality complements the other - while sensors offer a global view of the activity in the indoor space, cameras when integrated with the sensors can provide more specific information to identify persons or activities. The same functionality cannot be achieved merely using sensors and would be extremely difficult using just cameras.

In most real-world surveillance scenarios, one is confronted with the problem of allocating resources (e.g. cameras) – that are limited and expensive – to several competing resource requests (for example observing humans). As already discussed, deploying a dense camera network is not a practically feasible solution to the problem. Therefore, the real-world challenge is to gather as much information about humans and their activities with as few resources as possible. Traditional approaches which involve tracking of humans in video and analysis of activities using single or multiple Pan-Tilt-Zoom (PTZ) cameras are typically based on the assumption that the cameras can be *committed* to that particular task or *hypothesis*. This can lead to several drawbacks and limitations – committing to a particular hypothesis necessarily means that other possible hypotheses are discounted and any future evidence supporting them is ignored. Due to the dynamic environments in which surveillance systems are deployed, it is more of a rule than an exception to find a previously discounted hypothesis turning into a leading one later on. Practically, this means that due to the limited field-of-view of the cameras, it is impossible to recover lost information – such as identities of humans and activities – that is not visually observed due to the camera being committed elsewhere. Thus, this requires a compromise between committing the expensive resource (in our case, the cameras) for a task as limited in scope as tracking a particular entity/person or analyzing its behavior and maintaining all possible hypotheses till a decision is absolutely necessary. Hence, this requires a) strategies that maximize the camera’s capabilities by following a *minimal commitment* policy, and b) alternate modalities of sensing that provide useful information about discounted hypotheses even when the cameras have been committed to a specific one.

**Overall Approach:** Monitoring an indoor space with many humans with a small set of cameras is a very difficult and even

P. Turaga is with the Center for Automation Research, University of Maryland Institute for Advanced Computer Studies, College Park MD 20770 USA e-mail: (pturaga@umiacs.umd.edu). Y. Ivanov is with Mitsubishi Electric Research Labs, Cambridge, MA 02139 USA email:(yivanov@merl.com)

ill-posed problem. In this situation, inexpensive sensors that can be easily sprinkled throughout the building provide crucial context and global information about emergent behaviors. Designing a real-time system that exploits the strengths of sensors and cameras brings up interesting challenges. The primary problem is to achieve seamless integration of the different sensing modalities. We follow the principle of responding to events as they happen, where they happen. This requires development of approaches which fuse sensor data with camera control strategies and machine learning/computer vision algorithms. We exploit motion sensors to set primitive alarms in the surveillance system that detect simple human movement patterns. We further show how these primitives can be used to construct models for more complex activities that can be parsed on-line. Cameras are integrated into this on-line activity analysis system to provide snapshots of persons involved in the activities of interest. We use computer vision algorithms to detect and recognize humans from these snapshots. Finally, we show how the sensor network can be exploited to deploy a camera scheduler to provide shots of people walking in the common areas of the indoor space. The rest of the paper is devoted to discussing how we achieve these goals in more detail. Finally, the design of an interface that presents the maximum amount of information to a user in an easy and intuitive fashion is also relevant. We stress here that the purpose of the paper is not to compare recognition results of standard computer vision or machine learning methods that address some of the above tasks. The broader pursuit is to design a real-time system that can serve as a platform to assist and deploy more sophisticated algorithms from computer vision and machine learning so they can more effectively tackle the issues encountered in surveillance settings as discussed above.

**Contributions:** The following are the major contributions of this paper.

- 1) **Surveillance strategy:** We propose a novel strategy involving minimally committed cameras and ambient sensors to perform surveillance tasks. This approach maximizes information content and allows efficient recovery from potentially misleading decisions.
- 2) **Multimodal Integration:** We propose a system to efficiently integrate two widely different modalities – motion sensors and cameras – for scalable real-time surveillance.
- 3) **Gestural Interface:** We present a novel solution to the problem of visualization and interaction with the multimodal data that conveys maximum information pre-attentively.

**Organization of Paper** In section II, we provide an overview of the surveillance scenario and the proposed approach and briefly discuss the principles that guided us in designing the user interface for our system. In section III, we show how we exploit the sensor network and cameras to performing various surveillance tasks such as event detection, human and/or object detection and active scheduling. In section IV, we demonstrate the strength and generalizability of the proposed approach for several surveillance tasks. Finally, in section V, we present concluding remarks and directions

for future research.

## II. SYSTEM OVERVIEW

From the previous discussion, we now enumerate the principles that guided us in our pursuit of a real-time multi-modal surveillance system.

- **Maximizing resource use:** Maximizing the usage of limited camera resources without loss of potentially useful information.
- **Ambient Information:** Exploiting ambient information using motion sensors to extract information and assist cameras in surveillance tasks.
- **Scalability and Extendibility:** Designing a system that is easily scalable both in size of the observed space and density of visual and motion sensors and easily extensible by allowing deployment of more sophisticated algorithms for analysis of events etc.

Following these guidelines and requirements, we give a brief overview of the proposed multi-modal integration. A schematic of the on-line system is shown in figure 1. As is shown in the figure, an indoor space is augmented with a ubiquitous sensor network and a few pan-tilt-zoom cameras. Together the cameras and sensors form the physical elements of our system, which interacts with these physical elements via the following modules, as shown in figure 1:

- 1) **Camera Controller:** The camera controller is the interface between cameras and the rest of the software modules. It manages the cameras by passing commands that control the pan-tilt-zoom parameters of the camera.
- 2) **Activity and Event Analysis:** Events of interest either specified a priori or event models specified on-line by a user are automatically detected and corresponding visual evidence of the event is further processed. This module recognizes events of interest by utilizing the ubiquitous sensor data to continuously parse for event models and send appropriate messages to the camera controller to gather visual evidence of the activity.
- 3) **Image Analysis:** Images gathered from the cameras in response to event detection or sensor activations are analyzed using computer vision algorithms. Image analysis methods are used to perform tasks such as localizing humans for biometrics such as human ID, extracting information such as size and location of humans on the image-plane, which is then used to foveate cameras to the exact location of the face or human for capturing higher quality images etc.
- 4) **Scheduler:** Maximizing the limited field-of-view (FOV) capabilities of the camera and keeping in line with the minimal commitment philosophy requires active and optimal control of the cameras to gather visual imagery that is as informative as possible about the indoor space and its occupants. The data from the sensor network is used by the ‘scheduler’ to focus the camera toward the physical location of sensor activations and capture useful visual evidence of who or what is causing the activation by optimally allocating the cameras to the various sensor activations at any given point of time.

- 5) Decision Module: The Decision module mediates between the competing requests for camera resources by the scheduler, event analysis and the image analysis modules.

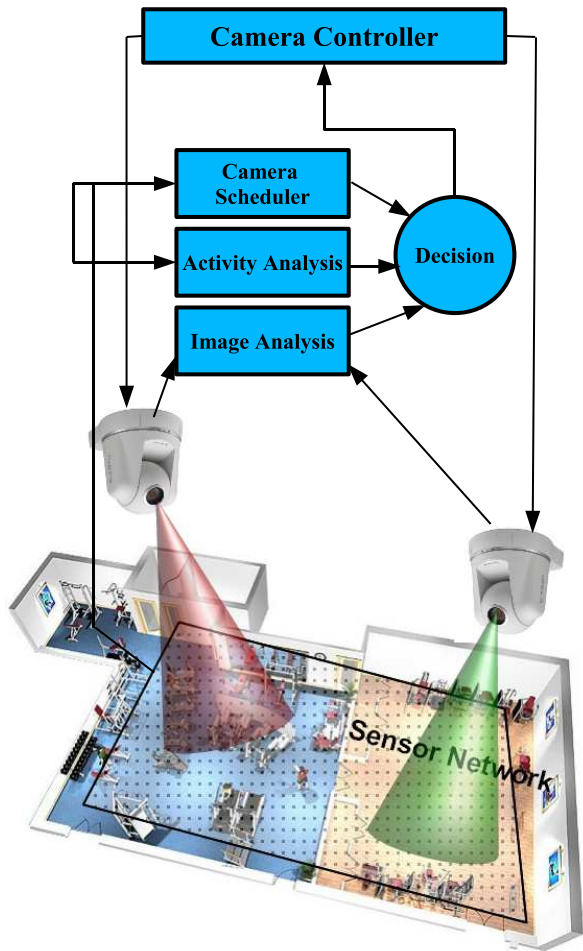


Fig. 1. Overview of real-time integration of sensors, cameras and control and detection algorithms.

Each of the modules is described in greater detail in subsequent sections. Prior to that, we first formalize the notions of an indoor-space that is augmented with a few cameras and a few sensors. Then, we discuss the representation of events in this framework.

#### A. Problem formulation

An indoor-space  $\mathbb{S}$  is a 3D space occupied by humans. Equivalently, we can consider it to be a 2D map of the 3D space seen from the top-view as shown in figure 2. Thus,  $\mathbb{S} \subset \mathbb{R}^2$ . The indoor-space  $\mathbb{S}$  is augmented with  $M$  cameras  $\mathbb{C} = \{C_i\}_{i=1}^M$ . We define the *coverage* of a camera as the subset of  $\mathbb{S}$  that is directly observable by the camera i.e.  $cov(C_i) \subset \mathbb{S} \forall i$ . Denote  $\mathbb{U} = \mathbb{S} - \bigcup_i cov(C_i)$  as the unobservable areas of  $\mathbb{S}$ . If  $\mathbb{U} = \phi$ , then the space is ‘fully covered’ or fully observable. If  $cov(C_i) \cap cov(C_j) = \phi, i \neq j$ , then the cameras have mutually exclusive FOVs. In our case  $\mathbb{U} \neq \phi$  and the cameras may have overlapping FOVs. Now given a point  $P \in \mathbb{S}$  where  $P = (x, y)$ , we define the *visibility* of  $P$  as the set of cameras that can observe it. i.e.  $vis(P) = \{C_i | P \in cov(C_i)\}$ . In

general, given a point  $P$ ,  $vis(P)$  may contain a single camera or many cameras or even none.

In the current situation, the cameras are dynamic PTZ cameras. Since the PTZ parameters are subject to change, we define the state of a camera  $C_i$  at time  $t$  as its current PTZ parameters  $S_i(t) = (p, t, z)$ . Note that the coverage of a camera includes all those spatial points that can be seen by varying its PTZ parameters over its full range.

Now we discuss how the motion sensors come into play. Motion-sensors are used to detect occupancy in a small spatial neighborhood. Thus, instead of having to deal with the whole space  $\mathbb{S}$  we now only deal with the *quantized* version of the space  $\mathbb{S}$  i.e. we partition  $\mathbb{S}$  into non-overlapping rectangular regions, where the size of the region corresponds to the coverage area of the sensor. With this in mind, we now consider coverage of a camera as the set of sensors that can be observed by the camera. Similarly, visibility of a sensor is the set of cameras that can observe it.

We consider events as spatio-temporal patterns of human behavior. Primitive events such as someone entering or exiting the office are considered to be localized in both space and time. Thus, we can associate a space-time location  $(x, y, t)$  to a primitive event  $E_p$  and ask questions such as which cameras can observe  $E_p$  or which camera requires the *smallest* change in parameters to observe the event etc. On the other hand, complex events such as a meeting in the conference room or someone committing a robbery are considered as sequences of primitive events and thus extend both in space and time.

#### B. Motion Sensors

In our system we use a network of wireless motion sensors of our own design, which was presented elsewhere (e.g. see [1]). Different implementations of this basic sensor design are inexpensive and consume very little power in their operation. For instance, second and third generation of the sensor used in [1] last up to 3 years on a single 3.3V battery, thereby dramatically reducing installation cost. In the system used in this experiments a total cost of the 155 sensors nodes is roughly equal to half the cost of the 6 video cameras in the installation. In the interest of privacy, the sensors are only installed in hallways and other public spaces, such as reception area, conference rooms, lunch room and kitchen.

#### C. Camera Sensor Calibration: Coverage and Visibility

Our system consists of a network of 155 motion sensors and 6 PTZ cameras deployed in a 3000 m<sup>2</sup> office space which is occupied by about 80 people. The sensor network in our system consists of individual wireless sensor nodes. Motion sensors cover most of the common areas. Each sensor has a FOV of about 4 m<sup>2</sup>. Each node sends a signal to the central server once a motion is detected in its field of view. We refer the interested reader to [2] for further details of the sensor network hardware.

During installation, the location of each sensor in the global coordinate frame is recorded and stored in an SQL database. This information allows us to simply locate each sensor on the map panel of the user interface, which also depicts the floor

plan and locations of the cameras. After the sensor locations are determined, we perform a simple calibration of the sensor network and PTZ parameters of each camera. This process allows us to calculate, for each sensor, the range of PTZ parameters of each camera such that any parameter from the calculated range ensures that the area covered by the sensor is also observed by the camera. This information is stored in the SQL database. This calibration database now provides the mechanism for computing the coverage of each camera and the visibility of each sensor in the cameras.

#### D. Visualization and Interface

We have chosen ease of understanding and simplicity of use as the guiding principles to design the user interface. Snapshots of the interface are shown in figures 2 and 3. In the center of the interface is the floor-plan of the indoor space with demarcated cells to show the positions of the various sensors. This type of display is intuitive, and communicates a coarse version of what is currently going on at a global scale even in a fleeting glance. This panel is called a *map*. Images from the cameras are displayed as floating elements overlaid on the map at the locations where the cameras are physically located. These floating elements can be moved around and zoomed into using simple gestures.

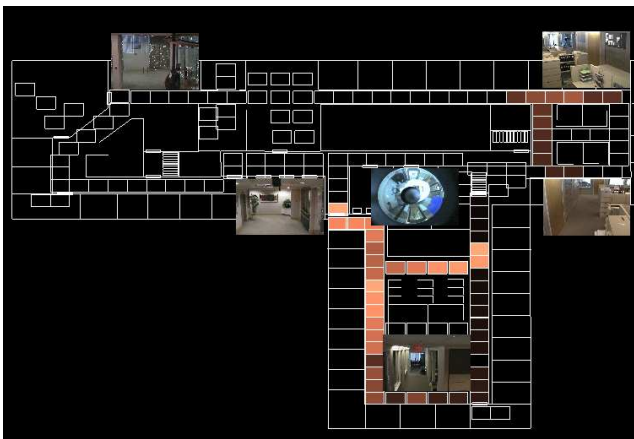


Fig. 2. The interface for the live system.

### III. SYSTEM ARCHITECTURE

In order to achieve the goals of on-line monitoring of our indoor space as formulated earlier in section I our system consists of three modules – a) Camera Scheduler, b) Event Analysis and c) Image Analysis. The Camera scheduler module is responsible for directing cameras towards the parts of the space, where it makes most sense for them to record. Event and Activity analysis module allows the user to formulate the ‘special cases’ and override the default scheduler behavior by specifying sets of events which cause cameras to foveate to the corresponding locations. And finally, the Image Analysis module allows automatic detection of people with the purpose of automating the task of gathering the best face shots of passing people by zooming in on the areas of the image where a person’s face is most likely to be.

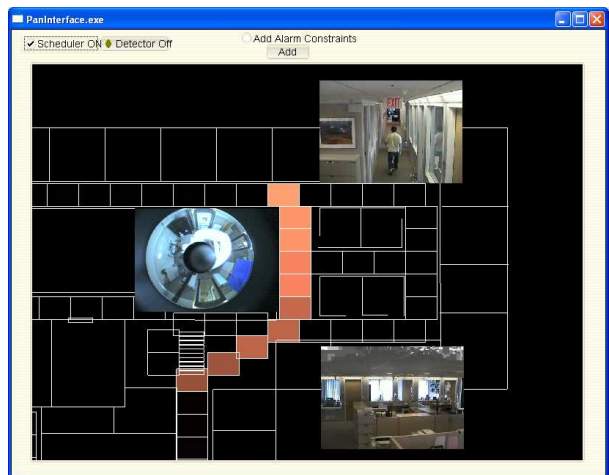


Fig. 3. A portion of the map zoomed in.

#### A. The Camera Scheduler Module

With only a few cameras observing a large indoor space, it is not possible to observe all the occupants walking by in a passive surveillance system. It becomes necessary to actively control the cameras so that most of the people passing by are observed by the cameras. The need for this is also motivated from the minimal commitment requirement of the cameras as discussed in section I. We regard this problem as one of resource management and allocation. Most camera scheduling systems rely on a set of wide field-of-view stationary cameras and a few active PTZ cameras such as [3] and [4]. The stationary cameras are used to infer the ‘state-of-the-world’. The location of humans is estimated coarsely from the stationary cameras, then the PTZ cameras are used to capture higher resolution images. Our work differs significantly from such approaches by using the sensor networks capabilities. In our approach, the network of impoverished sensors maintains the state-of-the-world and informs the PTZ cameras of the physical locations of humans in the space. Since, the PTZ cameras are calibrated with respect to the physical space (described in section II-C), they can apply the appropriate PTZ parameter values needed to observe that space.

Data from sensors is regarded as a request for a resource - the resource being the PTZ camera. All incoming requests are maintained in the form of a priority queue. For each block of time of about 10ms, we list the sensors that are active in that time-window. The latest request is sent to the back of the list. Let  $A^{(t)}$  denote the list of sensor activations during the time-window centered at  $t$ . For each sensor in the activation list  $A_i^{(t)}$ , we first find its visibility set  $vis(A_i^{(t)})$ . In general, there will be more than one camera that can observe the corresponding physical location. For each ordered pair of sensor activation and camera, we can define a cost of allocation. If a camera is not in the visibility set of  $A_i^{(t)}$ , we assign an allocation cost of infinity. For cameras in the  $vis(A_i^{(t)})$ , we define allocation cost as the required change in the PTZ parameters to observe the sensor. To do this we need to compute the required change in the state of the camera. Let  $S_k^{(t)}$  be the current state of camera  $C_k \in vis(A_i^{(t)})$ .

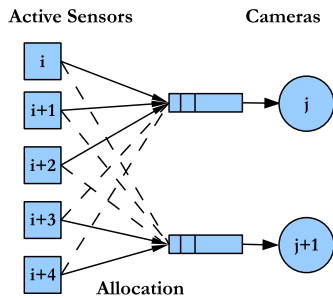


Fig. 4. Illustration of the scheduling process. The arriving requests at each camera are maintained as a queue at the cameras. Each sensor has multiple cameras that can potentially serve the request (shown as dotted lines), and one particular assignment as decided by the scheduler is shown as solid lines.

Let  $\hat{S}_k$  be the state required to observe the sensor (which is computed from the calibration database). Then the cost of allocation  $cost(A_i^{(t)}, C_k) = d(S_k^{(t)}, \hat{S}_k)$ , where  $d(\cdot)$  is some distance metric on the state-space of the cameras. In our case, the state of a camera was defined to be the current PTZ values i.e.  $S_k^{(t)} = (p, t, z)$ . In our implementations, the zoom parameter is not used by the scheduler. Instead, it is used by the image-analysis module to get better face-shots of people as will be described in section III-C. Hence, for the purposes of scheduling, we simply define  $d(\cdot)$  as the euclidean norm between the current and required pan-tilt values only. Thus, if  $S_k^{(t)} = (p, t)$  and the required PT values to observe the  $i^{th}$  activation  $A_i^{(t)}$  is  $\hat{S}_k = (\hat{p}, \hat{t})$ , then  $cost(A_i^{(t)}, C_k) = \sqrt{(p - \hat{p})^2 + (t - \hat{t})^2}$ .

In general there will be more activations than the number of cameras. This requires cameras to quickly process high-priority requests and move on to the next ones. Figure 4 is a simplified illustration of the process. In our system, the scheduler starts from the top of the queue of sensor activations and assigns to each request a single camera with the least allocation cost. Requests that could not be serviced within a preset time-lapse are removed from the queue. Though suboptimal and fast, we found that this works well in many real experiments.

### B. The Event and Activity Analysis Module

Representation and recognition of human activities from video is a well researched area in computer vision. A good survey can be found in [5]. Probabilistic approaches such as Dynamic Belief Networks [6], Hidden Markov Models (HMMs) [7], syntactic approaches such as Stochastic Context Free Grammars (SCFG) [8], Petri-nets [9] and others, have been employed for the task of recognizing activities.

General human activities in indoor spaces can be very complex. They often involve a number of people and objects and usually imply some form of temporal sequencing. For example, consider the activity of two people meeting in the lounge of a large office space and exchanging an object, say a briefcase. This simple activity is composed of several primitives - a) two people enter the lounge independently, b) they stop close to each other, c) the briefcase is transferred from one person to the other, and d) the persons leave. The activity starts with two independent movements which

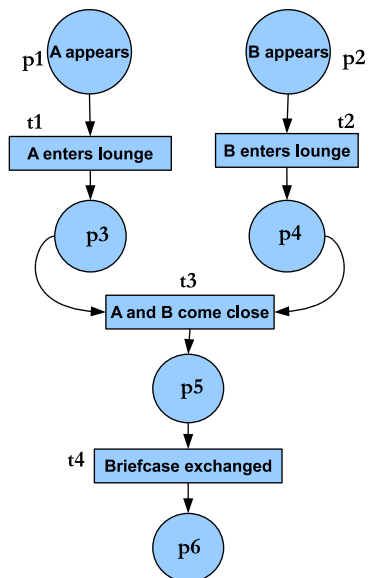


Fig. 5. Example of briefcase exchange by two people.

happen concurrently. The movements come to the temporal synchronization point, at which time the suitcase is exchanged, and then diverge again into two independent motions as people leave the room. Such situations where observations form independent streams coming into synchrony at discrete points in time are conveniently modeled using Petri-nets.

1) *Petri-nets*: Petri-nets were first introduced by Carl Adam Petri [10] as a mathematical tool for describing relations between conditions and events. They are particularly useful to model and analyze behaviors such as concurrency, synchronization and resource sharing, and traditionally found wide use in the theory of operating systems and compiler design. We refer the reader to [11] and [12] for a good survey. Formally, a Petri net is defined as follows:

$$PN = \{P, T, \rightarrow\}$$

where

- 1)  $P$  and  $T$  are finite disjoint sets of places and transitions respectively i.e.  $P \cap T = \emptyset$ .
- 2)  $\rightarrow$  is the flow relation between places and transitions, i.e.,  $\rightarrow \subseteq (P \times T) \cup (T \times P)$ .
- 3)  $\{p \in P \mid \exists t \in T \text{ s.t. } p \rightarrow t\} \neq \emptyset$ . This means that there exists at least one end place in the Petri net.
- 4)  $\{p \in P \mid \exists t \in T \text{ s.t. } t \rightarrow p\} \neq \emptyset$ . This means that there exists at least one start place in the Petri net.

Further,

- 1) The preset of a node  $x \in P \cup T$  is the set  $\{y \mid y \rightarrow x\}$ . This set is denoted by the symbol  $\cdot x$ .
- 2) The postset of a node  $x \in P \cup T$  is the set  $\{y \mid x \rightarrow y\}$ . This set is denoted by the symbol  $x \cdot$ .

The Petri-net representation of the aforementioned briefcase exchange example is shown in figure 5. This example illustrates concurrency, synchronization and sequencing.

2) *Petri-net Parsing*: The dynamics of a Petri-net (PN) are represented by *markings*. A marking is an assignment of *tokens* to the places of the Petri-net. The execution of a Petri-net is

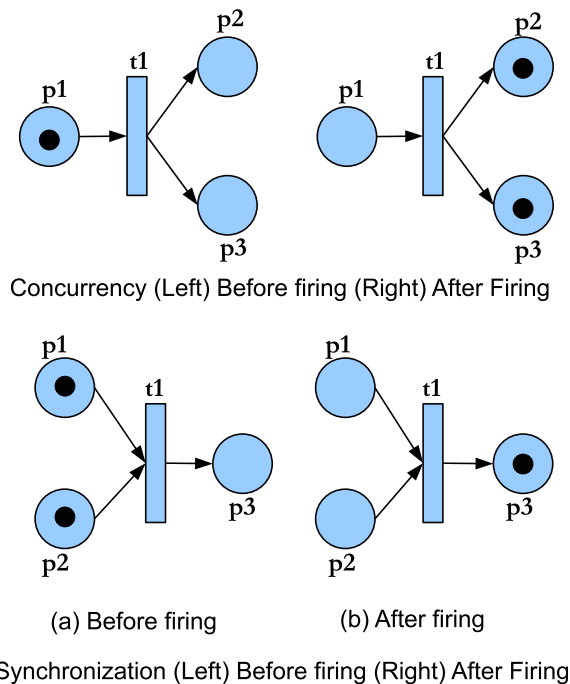


Fig. 6. Illustration of Petri-net firing for the case of concurrency and synchronization.

controlled by its current marking. A transition is said to be enabled if and only if all its input places (the preset) have a token. When a transition is enabled, it may *fire*. In the simplest case, all the enabled transitions may fire. We can also associate other conditions to be satisfied before an enabled transition can fire. When a transition fires, all enabling tokens are removed and a token is placed in each of the output places of the transition (the postset). Illustration of the firing process is shown in Figure 6 corresponding to the cases of *concurrency* and *synchronization*.

**Example 1:** For the briefcase example shown in figure 5, the places are labeled  $p_1, \dots, p_6$  and transitions  $t_1, \dots, t_4$ . In this PN,  $p_1$  and  $p_2$  are the start places and  $p_6$  is the end place. When a person enters the scene a token is placed in place  $p_1$ . In this state, the transition  $t_1$  is enabled. But it does not fire until the condition associated with it is satisfied i.e. the person should enter the office lounge. Once this happens, the token is removed from  $p_1$  and placed in  $p_3$ . Similarly, when another person enters the scene a token is placed in  $p_2$  and transition  $t_2$  fires after the person enters the lounge. Now, with a token in each of the enabling places of transition  $t_3$ , it is ready to fire when the associated condition occurs i.e. when the two people come close to each other. Then,  $t_3$  fires and both the tokens are removed and a token is placed in  $p_5$ . Now  $t_4$  is enabled and ready to fire. It fires when the briefcase is exchanged between the two people. After it fires, the token is removed from  $p_5$  and placed in  $p_6$  which is the end place. When the token reaches the final place, the activity is completed.

**Application in the System:** The Petri-net formalism provides a rich tool for representing and recognizing events from time-series data. For practical application, we need a) to define the primitives for the events, b) provide the method to extract these primitives from data, and c) define the semantics

of the event. In our system, we use the above formalism as a guiding force. The primitives for the activities are the simple human movement patterns which are detected using the sensor network. We provide an interface to the user to build models for activities using these primitives. We describe the process of activity specification as well as the activity detection algorithms in the subsequent sections.

3) *Detecting Primitive Actions and Complex Activities:* A basic requirement of an on-line surveillance system is the ability to set ‘live alarms’ on the fly. Live alarms allow a user to capture visual evidence of activities of interest. The alarms could correspond to abnormal activities such as someone entering a bank safe or as an intermediate step toward performing some other task such as counting the number of people who access the printer-room in a day. We rely on the sensor networks capabilities to set these primitive alarms due to the relative ease of parsing sensor activations and the robustness of the sensors. These primitives usually correspond to a sequence of sensor activations. The sequence of activations can be specified by the user by *tracing* a path of interest on the map. The alarm ‘goes off’ whenever the specified sequence of activations occurs.

The primitive alarms are modeled as finite state machines (FSM), where each sensor acts as a state and the specified sequence as the overall FSM. For every incoming sensor data, all the available FSM’s are continuously parsed. When an alarm goes off, a message is sent to the camera control module to foveate the cameras toward the physical location of the alarm. Once the cameras foveate to the appropriate physical location, visual evidence of the activity at the scene is captured for further image analysis.

In certain settings, one can enumerate these ‘interesting’ activities and design hand-crafted models for them. But in most cases, it cannot be judged a priori what events a user might consider watch-worthy. Hence, it is a desirable feature to allow the user input a model for an activity of interest, and let the system parse that model on-line. But, this comes at a price - in general, models for activities can be arbitrarily complex. The ability to input arbitrarily complex models would compromise the simplicity and intuitive nature of the interface. While, in principle, we could have a separate interface to deal with the complexity, we limit ourselves to simple conjunctions of primitive alarms. A small set of conjunction operators such as ‘AND’, ‘OR’, ‘AFTER’, ‘BEFORE’ can be mapped to simple gestures without the need for a separate interface. Even this limited set of conjunction operators adds a significant level of sophistication to the activity models.

### C. The Image Analysis Module

One of the most salient applications of computer vision in security and surveillance is face recognition. Traditional approaches include feature based methods [13] and ‘global’ methods such as PCA [14]. This was followed by other pattern recognition approaches including linear discriminant techniques [15], neural network based algorithms [16] and methods based on graphical models [17]. Newer computational tools from machine learning such as SVMs [18] and boosting

[19] have been successfully applied to this problem in recent years. Variabilities such as resolution, pose and illumination are the main limiting factors for practical application of these algorithms. Further, in unconstrained settings one also needs to deal with the effects of occlusion. We differ from these approaches by exploiting sensor network capabilities to counter some of these limitations and improve performance.

Face recognition is well understood when the image of the person’s face has high resolution and canonical pose. Unfortunately, in most security systems, cameras are mounted just under the ceiling and provide a poor view of a person’s face. The goal of the Image Analysis module is to allow us to simplify the collection of good views of faces of people passing through the space. To minimize the effect of the ceiling mounted camera on pose and to ensure that camera’s record faces and not backs of people’s heads, we specify two types of rules - foveation and zoom. Foveation rules are those that direct cameras to foveate to objects moving towards them to make it more likely that the person’s face is recorded. The zoom rules specify zoom parameters to the foveated camera if a person is detected in the camera view such that the most likely face area is maximized.

**Zoom Rules:** The relationship between zoom value and object-size is non-trivial. Instead of modeling this relationship explicitly, we assume a linear relationship between zoom parameter and object size, which works fairly well in our case. Using the results of the face/pedestrian detector, we deploy a ‘human-zoomer’ module that zooms into the objects to provide higher quality images. Given the current size of the object (face/pedestrian) in the image, we control the zoom parameter of the camera to get the object size to a predefined level.

**Foveation Rules:** A foveation rule is specified by setting alarms corresponding to paths leading to the camera. Such alarms will ensure that the cameras capture faces in the frontal pose, making the task easier for face detection and recognition algorithms. This results in improved accuracy than relying purely on image analysis.

#### IV. EXPERIMENTS

In this section, we present a few experiments that demonstrate the flexibility of the proposed approach and power of the multi-modal integration. All tests were performed in unconstrained real-world conditions – the indoor space was the Mitsubishi Electric Research Lab which is occupied by about 80 researchers. The experiments and tests demonstrate the effectiveness and generalizability of the approach for several surveillance tasks.

It will always remain a challenge to specify what type of data should be considered a “ground truth” for this kind of system for its quantitative evaluation. In our deployment, we have tried to optimize the use of the available cameras by maximizing the area of coverage. In spite of this, several parts of the office-space are not visually observed. Since we rely on the sensors to detect primitive and complex activities, it is difficult to collect ground truth unless a camera observes the specific location of the activity. Due to these constraints, we show sample activity detections with corresponding visual evidence when available.

##### A. Event Detection: Live Alarms

Live alarms are modeled as finite state machines, where each sensor acts as a state and the specified sequence as the overall FSM. Live alarms are set by a user by drawing paths of interest on the map panel. For every incoming sensor data, all the available FSM’s are parsed. When an alarm goes off, a message is sent to the camera control module to foveate the cameras toward the physical location of the alarm. Once the cameras foveate to the appropriate physical location, visual evidence of the activity at the scene is captured for further image analysis. Figure 7(a) shows an example of spatio-temporal alarms that are set by *tracing* paths on the map. Figure 7(b) shows one of the alarms going off. The cameras automatically foveate to the appropriate physical location on the map using calibration as described in section II-C.

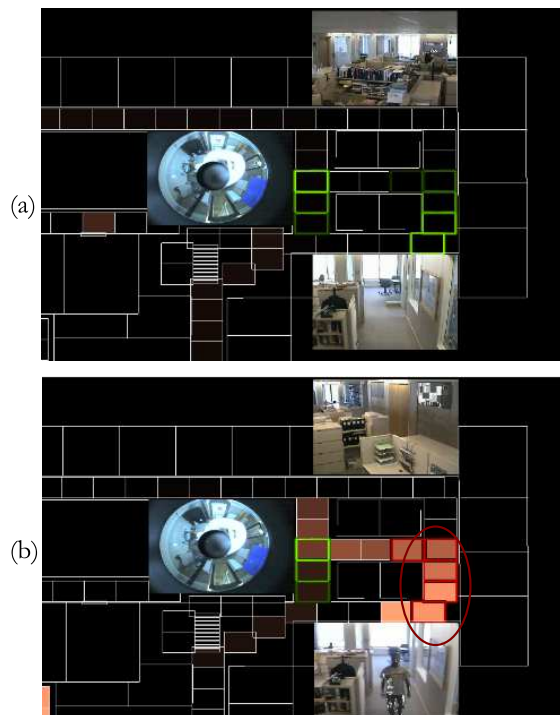


Fig. 7. (a) Spatial-temporal live alarms displayed in green, (b) Alarm that goes off is highlighted in red and camera foveates to the alarm location.

The alarms stay active until disabled by the user. Our system allows any number of alarms to be set simultaneously. From a visualization perspective, the active alarm sequences are highlighted in green. The sensors are highlighted with varying intensities - the first sensor has lowest intensity while the last one has the highest intensity - which corresponds to the ‘direction’ of the alarm. The alarms turn red when they go off and stay red with decreasing intensity representing the time-lapse since the alarm went off. After a pre-defined interval, the alarm is reset to green again. The transition from green to red proves useful to attract the attention of the user.

##### B. User-defined Complex Event Detection

In this experiment, we show how two interesting scenarios – sequencing and synchronization, are easily modeled using

the Petri-net formalism. In figure 8(a) we show a model of a person entering the office and then entering the kitchen which corresponds to a sequencing constraint. The sequence consists of two primitive actions `Person enters office`  $\rightarrow$  `Person enters kitchen`. Each of these primitives is specified as a live alarm. The sequencing constraint between them is specified by the user using the gestural interface. Temporal constraints such as the temporal duration of each action primitive and the time-lapse between action primitives can be input as well. Figure 8(b) shows an example of the activity and the reported detection. Since no camera visually observed the kitchen, no visual evidence is presented.

Similarly, in figure 9(a) a model of a meeting near the director’s office is provided by the user. This activity is characterized by two separate threads which have a particular spatio-temporal synchronization point. Figure 9(b) shows an example of the activity. The corresponding images show the persons involved in the meeting. Note the change in camera position once the activity constraints are satisfied.

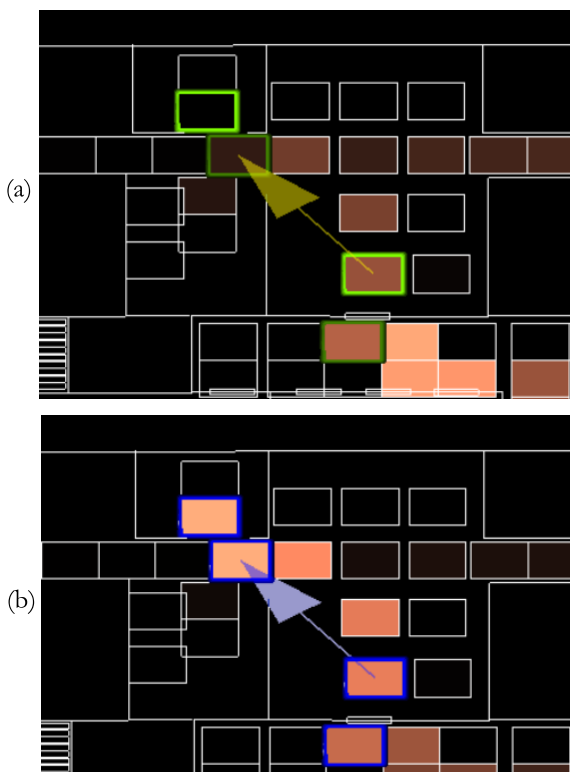


Fig. 8. (a) Live alarms linked by sequencing constraint, (b) Alarms go off when the constraints are met.

### C. Event-driven Camera Scheduling

As described in section III-A, cameras can be actively controlled to provide visual evidence of persons or activity using the proposed multi-modal linking strategy (section II-C). For the case of the camera scheduling, one is interested in capturing images of humans walking by in the indoor space. This can be seen as an event driven scheduling, where the event in question is the activation of any motion sensor in the office space. In our implementation, the scheduler assigns to each request (i.e. sensor activation) a single camera with

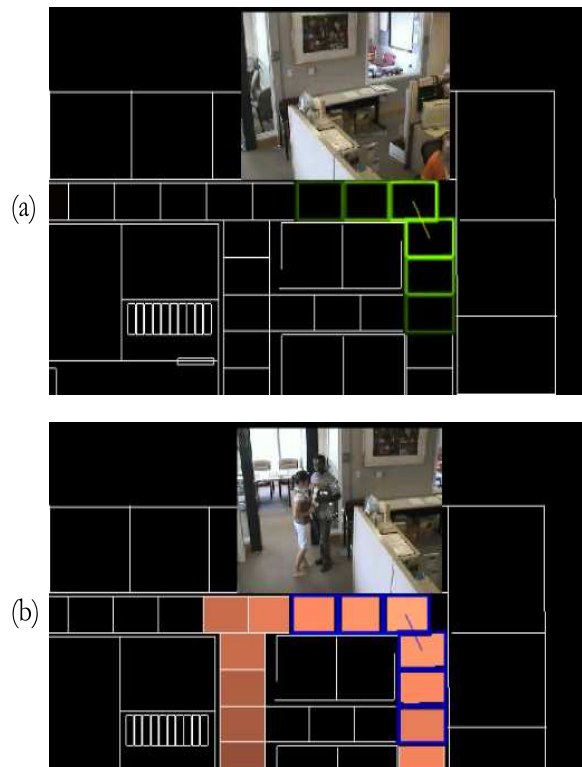


Fig. 9. (a) Meeting alarm set by simultaneous convergent sequences, (b) Meeting detected when constraints are satisfied.

the least allocation cost. Requests that could not be serviced within a preset time-lapse are removed from the queue. In figure 10, we show sample images captured by four cameras which are placed at different locations in the office space. We see that the cameras are assisted by the sensor activations to accurately foveate to the appropriate physical locations in the indoor space.

### D. Event-driven Image Analysis

So far, we have discussed how impoverished sensors are leveraged to set primitive alarms and specify activities of interest. We can now integrate this activity detection and alarm system with the PTZ cameras to further analyze the visual evidence that is gathered of the activity. At this stage, any of a wide variety of computer vision algorithms can be deployed for various tasks such as biometrics, behavior analysis etc. As a baseline, we implemented a system that relies purely on imagery to detect people in the camera’s FOV and zoom into them for capturing better resolution images. We realized that such a system fails very quickly (even if the false positive rate is low) due to the extremely cluttered environment and omnipresent occlusions in indoor spaces. In such a scenario, the sensors provide context for the image analysis algorithms. For example, instead of searching all over the PTZ parameter values and all over the image plane for, say, a human, the camera can wait for the sensor network to inform it of some activity, foveate to the corresponding physical location and then use image analysis algorithms. This can be achieved, for instance, by setting live alarms that correspond to people walking towards a camera. This not only

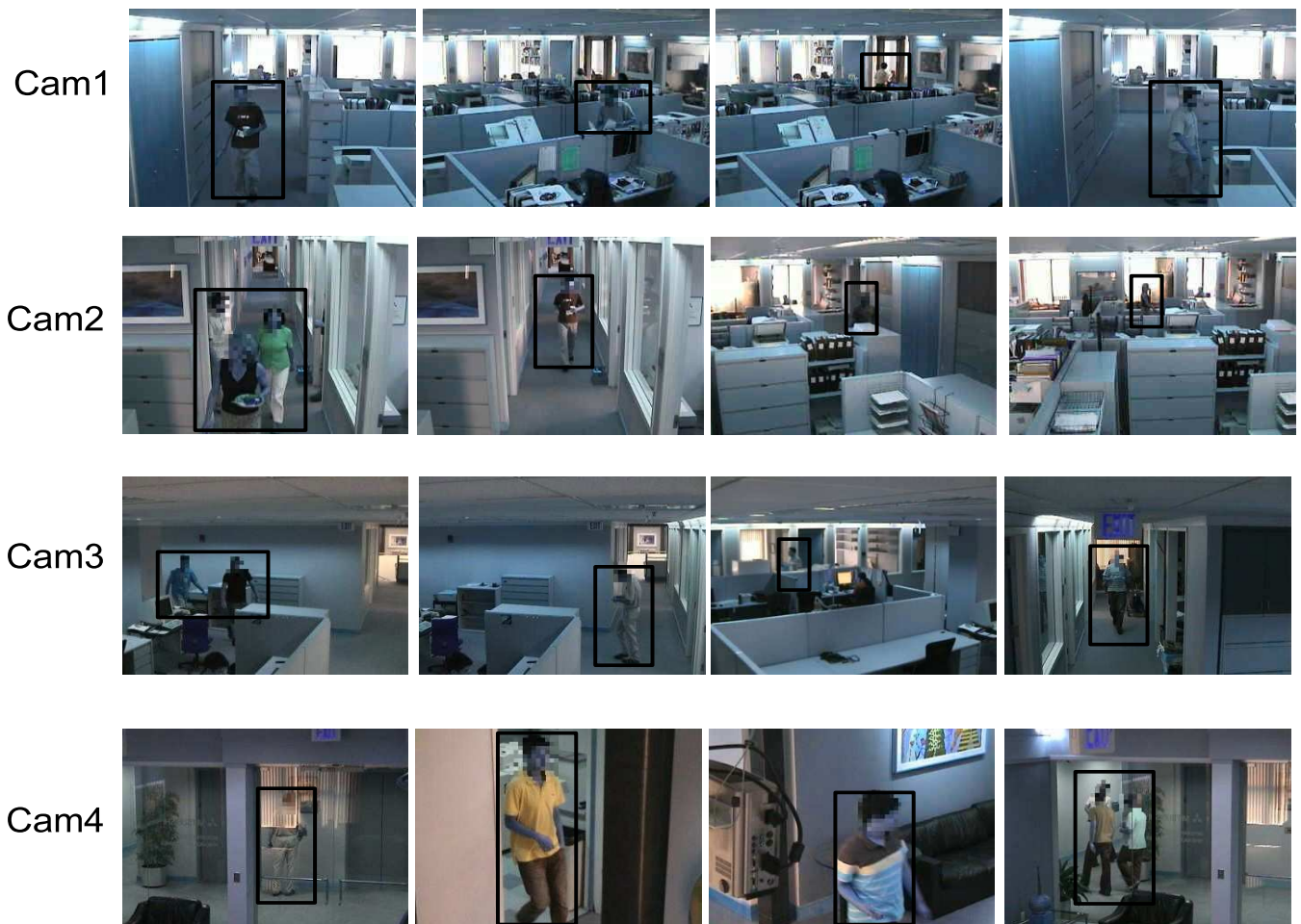


Fig. 10. Images captured by sensor based active scheduling of cameras. Scheduler exploits sensor activations to accurately foveate to the correct physical locations. Faces have been blurred to preserve privacy. Images have been manually annotated to show the locations of humans in the scene. Note that the scheduler only controls pan and tilt and not the zoom.

removes a huge computational overhead, but also improves the overall accuracy of the image-analysis tools. In our system, we have incorporated the well-known Viola-Jones detector [19] to detect faces and/or humans. We have used the OpenCV implementation. The detector easily runs at about 10 frames per second on images of size 320x240. The detector returns the  $(x, y)$  location, width and height of the bounding box of the detected objects on the image plane. We show some example detections in figure 11.

#### E. Multi-modal Forensics and Tracklets: Stringing Evidence

A method for multimodal tracking using multiple cameras using the framework of tracklet graphs was reported by [20]. We refer the interested reader to [20] for algorithmic details. The important improvement that the current approach offers over this earlier work is as follows. It is evident that sensor based active scheduling of cameras plays a major role in successful stringing of forensic video evidence. Minimally committed and event driven cameras maximize the information gathered from the limited amount of available resources resulting in higher quality results in forensic search and retrieval operations. Further, some of the tasks which were human guided can be replaced by automated image analysis algorithms such

as shape and appearance based matching of the tracked person across different cameras. However, in the absence of active scheduling, such automated matching techniques are prone to error and missed detections. Scheduling serves also to improve the performance of such image analysis algorithms. However, we do not explore that avenue in this experiment.

#### V. DISCUSSION AND FUTURE WORK

In this paper, we have presented a novel approach for real time surveillance with multiple modalities. We have presented our integration of two widely different modalities – motion sensors and cameras – and shown how we leverage the strengths of each. We have built a system that combines machine learning algorithms for activity recognition, image analysis algorithms for pedestrian/face detection with camera control strategies for efficient, real-time monitoring.

The general difficulty of evaluating the approach proposed in this paper is related to the sheer scale of the application. The main challenge of our application is to maximize the video coverage of the space and record the activity in a real office space. It is infeasible to install enough video cameras to provide such coverage to compare video annotations with the ones derived from our system. To address this problem we



Fig. 11. Example of Event driven People detection.

performed some activities (generally considered challenging in the literature) that were recorded in the database, and developed the algorithm that finds all instances of these activities in the database. While all activities that we staged were found, we do not know how many similar activities that were not staged were missed by our system. Finding a proper quantitative evaluation technique for similar situations remains a challenging task.

The activities performed in our system were challenging for any general purpose system. The challenge is three-fold. First, the specification of the detectors responsible for triggering the camera foveation needs to be extremely flexible, as at the manufacture and installation time, it might not be possible to know what kinds of activities will need to be detected. Thus, we focused our work on an approach that would let us build such detectors “on-the-fly”, without pre-coding any patterns. Second, much work has been focusing on detecting an activity of a single agent, agents taking turns, or interacting sequentially. Very little work exists in computer vision that addresses more realistic settings for multi-agent interactions - simultaneous coordinated activities. While solutions exist with specifying such activities with Petri-nets, these detectors are usually built prior to the commencement of the detection task. In our system multi-agent simultaneous coordinated activity patterns can be specified during its live operation, without expensive pre-training. And, finally, the scale of the application is its own separate challenge. Detecting of multiple events that may happen in disparate parts of an entire building with non-trivial geometry with video cameras alone might simply be infeasible. We proposed our solution that scales very well.

This approach opens up several interesting research prob-

lems. Can training examples of activities be extracted from video and used for learning models for normal and abnormal activities? Can we mine for activity patterns from recorded data. There are several future research directions that can be pursued. Automatic multi-camera tracking using computer vision algorithms is still an unsolved problem. Can it at least be used to make a human analyst’s job easier? We also plan to exploit video for recognizing activities. This is extremely challenging given the complex indoor environment and the PTZ capabilities of the cameras. We also hope to enhance the performance of standard vision algorithms for object detection by online learning where the sensors provide additional input.

## REFERENCES

- [1] C. R. Wren, Y. A. Ivanov, D. Leigh, and J. Westhues, “The merl motion detector dataset,” 2007 Workshop on Massive Datasets. Mitsubishi Electric Research Laboratories, Tech. Rep., 2007.
- [2] C. R. Wren and E. M. Tapia, “Toward scalable activity recognition for sensor networks,” *Location and Context Awareness*, pp. 168–185, 2006.
- [3] C. J. Costello, C. P. Diehl, A. Banerjee, and H. Fisher, “Scheduling an active camera to observe people,” *VSSN '04: Proceedings of the ACM 2nd international workshop on Video surveillance & sensor networks*, 2004.
- [4] F. Z. Qureshi and D. Terzopoulos, “Surveillance camera scheduling: A virtual vision approach,” *VSSN '05: Proceedings of the third ACM international workshop on Video surveillance & sensor networks*, 2005.
- [5] J. Aggarwal and Q. Cai, “Human Motion Analysis: A Review,” *Computer Vision and Image Understanding*, vol. 73, no. 3, pp. 428–440, 1999.
- [6] T. Huang, D. Koller, J. Malik, G. H. Ogasawara, B. Rao, S. J. Russell, and J. Weber, “Automatic Symbolic Traffic Scene Analysis Using Belief Networks,” *AAAI*, pp. 966–972, 1994.
- [7] M. Brand, N. Oliver, and A. Pentland, “Coupled hidden markov models for complex action recognition,” *CVPR*, 1997.
- [8] Y. Ivanov and A. F. Bobick, “Recognition of Visual Activities and Interactions by Stochastic Parsing,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 852–872, 2000.
- [9] C. Castel, L. Chaudron, and C. Tessier, “What is going on? A High-Level Interpretation of a Sequence of Images,” *ECCV Workshop on Conceptual Descriptions from Images*, 1996.
- [10] C. A. Petri, “Communication with automata,” *DTIC Research Report AD0630125*, 1966.
- [11] R. David and H. Alla, “Petri nets for Modeling of Dynamic Systems A Survey,” *Automatica*, vol. 30, no. 2, pp. 175–202, 1994.
- [12] T. Murata, “Petri nets: Properties, Analysis and Applications,” *Proceedings of the IEEE*, vol. 77, no. 4, pp. 541–580, 1989.
- [13] A. J. Goldstein, L. D. Harmon, and A. B. Lesk, “Identification of human faces,” *Proc. of IEEE*, 1971.
- [14] M. A. Turk and A. P. Pentland, “Face recognition using eigenfaces,” *CVPR*, 1991.
- [15] K. Etemad and R. Chellappa, “Discriminant analysis for recognition of human face images,” *Lecture Notes in Computer Science*, vol. 1206, 1997.
- [16] H. A. Rowley, S. Baluja, and T. Kanade, “Neural network-based face detection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 1, pp. 23–38, 1998.
- [17] L. Wiskott, J. M. Fellous, N. Krüger, and C. von der Malsburg, “Face recognition by elastic bunch graph matching,” *Proceedings of 7th International Conf. on Computer Analysis of Images and Patterns*, no. 1296, 1997.
- [18] E. Osuna, R. Freund, and F. Girosi, “Training support vector machines: An application to face detection,” *CVPR*, 1997.
- [19] P. Viola and M. Jones, “Rapid object detection using a boosted cascade of simple features,” *CVPR*, 2001.
- [20] Y. Ivanov, A. Sorokin, C. Wren, and I. Kaur, “Tracking people in mixed modality systems,” *Visual Communications and Image Processing*, 2007.