MITSUBISHI ELECTRIC RESEARCH LABORATORIES
http://www.merl.com

Tracking People in Mixed Modality Systems

Yuri Ivanov, Alexander Sorokin, Christopher Wren, Ishwinder Kaur

TR2007-011 February 2007

Abstract

In traditional surveillance systems tracking of objects is achieved by means of image and video processing. The disadvantages of such surveillance systems is that if an object needs to be tracked - it has to be observed by a video camera. However, geometries of indoor spaces typically require a large number of video cameras to provide the coverage necessary for robust operation of video-based tracking algorithms. Increased number of video streams increases the computational burden on the surveillance system in order to obtain robust tracking results. In this paper we present an approach to tracking in mixed modality systems, with a variety of sensors. The system described here includes over 200 motion sensors as well as 6 moving cameras. We track individuals in the entire space and across cameras using contextual information available from the motion sensors. Motion sensors allow us to almost instantaneously find plausible tracks in a very large volume of data, ranging in months, which for traditional video search approaches could be virtually impossible. We describe a method that allows us to evaluate when the tracking system is unreliable and present the data to a human operator for disambiguation.

VCIP 2007

Tracking People in Mixed Modality Systems

Yuri Ivanov[a], Alexander Sorokin[b], Cristopher Wren[a], Ishwinder Kaur[c]

[a]MERL, 201 Broadway, Cambridge, MA;
[b]UIUC, 201 N. Goodwin St., Urbana, IL;
[c]MIT, 77 Mass. Ave, Cambridge, MA

ABSTRACT

In traditional surveillance systems tracking of objects is achieved by means of image and video processing. The disadvantages of such surveillance systems is that if an object needs to be tracked - it has to be observed by a video camera. However, geometries of indoor spaces typically require a large number of video cameras to provide the coverage necessary for robust operation of video-based tracking algorithms. Increased number of video streams increases the computational burden on the surveillance system in order to obtain robust tracking results. In this paper we present an approach to tracking in mixed modality systems, with a variety of sensors. The system described here includes over 200 motion sensors as well as 6 moving cameras. We track individuals in the entire space and across cameras using contextual information available from the motion sensors. Motion sensors allow us to almost instantaneously find plausible tracks in a very large volume of data, ranging in months, which for traditional video search approaches could be virtually impossible. We describe a method that allows us to evaluate when the tracking system is unreliable and present the data to a human operator for disambiguation.

Keywords: Sensor networks, Surveillance, Tracking

1. INTRODUCTION

Growing availability of video cameras and cheap sensors make the technology of building large surveillance systems widely available. An indoor surveillance system may include a set of cameras and a set of sensors, access card readers and perhaps other sensing and registering equipment. An example of such setup can be seen in Figure 1. Each sensor individually may only cover a small portion of the space under surveillance, however, very little work is done on analysis and recognition of human behavior across different sensors and sensor modalities. That seems contradictory to how people typically use their spaces. A single camera is perhaps appropriate for such tasks as identification, or access control, where all necessary information can be contained in a single

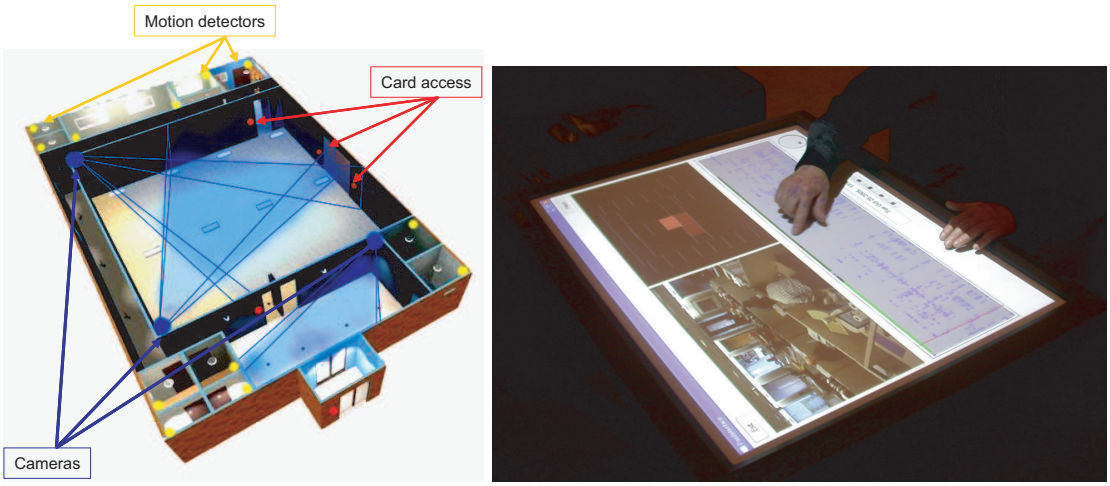

Figure 1. Left: Example indoors surveillance system configuration. Right: MERL search system, as a part of the DiamondTouch setup.

frame of video. In contrast, human behavior is typically extended in space and time and routinely crosses sensor boundaries. This paper and our system partially addresses these issues developing a space-centric mechanism for aggregating sensors and camera measurements within a single model.

Latest advances in technology of storage devices allow designers of video surveillance systems to further increase quality and capacity of the recording units. Increased storage capacity of modern digital video recorders (DVRs) enables corresponding increase in the number of video cameras used in such systems. For example, a Digital Video Recorder, released last year by Mitsubishi Electric Company[1] allows simultaneous recording of videos from 16 video cameras. Up to 16 of these devices can be easily "daisy-chained", so that 256 video feeds get stored in the array of storage devices totalling about 8 Terabytes in the most standard configuration.

Paradoxically, the increased storage capacity poses a challenge to the domain of video surveillance - a human operator or an automated system has to be able to handle such large volumes of video data in a very short period of time to provide a good degree of responsiveness of the system to search requests. To illustrate the point, if a security event can be localized to within a single hour, in the system with 256 video feeds may result in having to search through 256 hours of video data.

On the other end of the spectrum lie the problems of typical indoor surveillance, where the geometry of the space makes a complete video coverage prohibitively expensive. However, our earlier work[2] shows that a distributed network of simple motion sensors can be an inexpensive and accurate solution for a large number of surveillance and space automation tasks. We have also explored heterogeneous sensor network configurations that include sparse array of video cameras for typical tasks of forensic surveillance. We developed a system[3] that allows an operator to search a large database of sensor activations, arbitrarily joining them into search conditions with intuitive query interface. These queries, combined with flexible matching criteria and (at this stage) not using any computer vision algorithms permit fast retrieval of video segments that temporally coincide with the patterns of sensor activations, specified by the operator.

The topic of this paper is an introduction of a novel method for tracking individuals indoors in a typical office environment using the mixed modality sensor network described in our earlier paper.[3]

Most tracking algorithms have to address the issue of multiple objects present simultaneously in the camera view. This problem becomes even more pronounced when using ultra-low resolution sensors that are perceptually blind to the number of people in its view. However, constraints of larger spatial context may allow disambiguation of situations where multiple people are observed by a sensor network. For instance, when two people cross paths in the hallway a tracking algorithm can evaluate velocities of each person use these estimates to find a plausible interpretation of the sensor observations. However, this decision is based on a possibly incorrect assumption and once committed to, cannot be recovered from using the motion sensor data alone. In such situations the problem remains that for most practical applications of surveillance algorithms for general robust error-free tracking do not exist. Thus our general approach to the problems of this sort is to not focus on building a system that makes no mistakes, but rather a "self-critical" one, which makes explicit the degree of uncertainty about its own responses. This approach allows us to be efficient in the way it uses the human operator, making it possible to quickly narrow down the set of alternative interpretations of the scene by cohesively presenting to the operator only the information necessary for making the decision. In this paper we present our first experiences in implementing our vision in a full-scale sensor-camera network and its particular application to tracking individuals through an office space occupied by about 100 people. The remainder of this paper is organized as follows - section 2 briefly lists relevant to relevant work, section 3 describes the hardware implementation and architecture of the network, section 4 introduces the tracking extensions to the system introduced in our earlier work.[3] We conclude with section 5 giving brief discussion and future plans for the development of the tracking aspects of the system.

2. RELATED WORK

There is a significant body of literature surrounding the interpretation of human behavior in video.[4–8] A common thread in all of this work is that tracking is the very first stage of processing. That limits the work to sensor modalities that can provide highly accurate tracking information in the absence of any high-level inference. In particular, the ambiguities inherent in using a motion detector network can be expected to introduce enough noise in the tracking results to render most of these approaches unusable.

Figure 2. Left: A pan-tilt-zoom camera. Right: A wireless, passive, infrared motion detector.

There are a few works that have attempted to step outside this framework.[9,10] These systems learn task-specific state models that allow the behaviors to be recognized directly from the sensor data, without tracking. Our work follows this philosophy, and adapts it to the domain of sensor networks.

Wilson and Atkeson[11] also utilize a network of motion detectors. Their system is targeted at residences, where they assume that only a few individuals will be present. This allows them to pursue a classic track-then-interpret methodology. More people means more ambiguity, and more ambiguity means exponentially more hypotheses that must be considered during tracking. Therefore, this approach is only applicable to low-census buildings, such as homes. Wilson and Atkeson also assume strategic placement of sensors. That level of specialization is not economical in large buildings, or where usage patterns change regularly. We assume that our network will be built into the lights, outlets, and vents, and that it will likely be installed by professional electricians and ventilation engineers, rather than behavioral psychologists or eldercare specialists.

3. SYSTEM ARCHITECTURE

The system that we deployed at MERL consists of a network of wireless motion sensors and an six pan-tilt-zoom cameras (see Figure 2) affixed to the ceiling. The sensor network is built of sensor nodes that include passive infra-red (PIR) motion detectors. This is the same sensing technology used in most motion-activated lights and appliances on the market today. The sensor nodes are inexpensive, approximately $30 per node in prototype quantities of 1000, and much cheaper than that in full-scale mass production. The nodes are approximately 3cm by 5cm by 6cm.

3.1. Motion sensor nodes

The nodes use an industry-standard IEEE 802.15.4 radio. This is the physical layer typically used by Zigbee devices. These new radios are standards-compliant and therefore capable of inter-operating with equipment designed by many different vendors.

Measurements on a new sensor node show that it consumes approximately $50\mu A$ in 'motion detector' mode, and $46mA$ when communicating over the radio. However, communication is brief (16 ms). The node then returns to its 'motion detector' mode. Assuming that the sensor node detects motion every T seconds, this gives an average current consumption of

$$50\mu A + \frac{15ms}{T}46mA \tag{1}$$

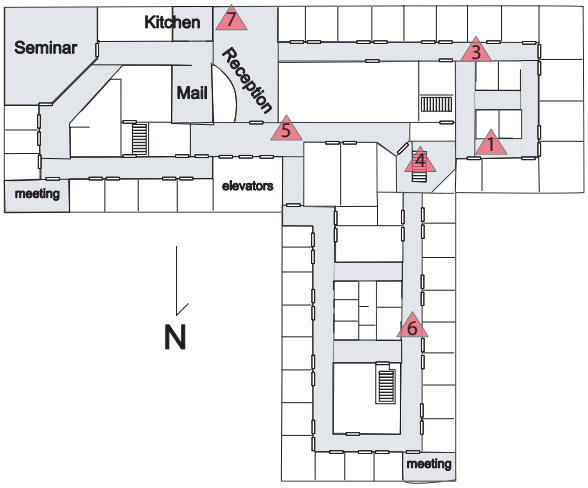

Figure 3. Map of the one floor of the office space. Shaded areas show public spaces where the sensors are installed. Locations of the six cameras are marked by small triangles.

Analysis of the current network reveals that the average inter-arrival time at a given node is 122 seconds. Even if a sensor is activated once per minute, it would still only draw an average current of $61.5\mu A$. A typical lithium AA cell with a capacity of $2Ah$ will therefore power our sensor for approximately 32,520 hours, longer than three years!

Long life on battery power (or parasitic power) is very important: it drastically reduces installation cost by eliminating the need for wires. Installation can become near-zero cost: just stick it up.

When motion is detected, a sensor-specific ID is broadcast over a wireless network. In our research prototype system, the packet is globally timestamped and copied to a conventional LAN for central storage and analysis. However, we anticipate that in a production system the nodes may communicate only locally by passing information directly between immediate neighbors to be analyzed, or to end-effectors in the system (such as an alarm system).

3.2. Sensor network in an office invironment

The map in Figure 3 depicts the test area. The network of 215 sensors covers 3000 square meters of office space occupied by over 100 people. Over nine months the system has recorded over ten million motion observations. Inter-arrival times statistics from this data were used in the design of the new nodes, as described in Section 3.1.

Executives and administrators occupy the wing on the right of the map. Researchers occupy the bottom and left wings. The central core of the building contains restrooms, lobbies, elevators, the mail room and the kitchen. In the center of the building there is a stairwell to another floor of the company that is mostly occupied by researchers.

4. TRACKING PEOPLE IN THE MERL MIXED MODALITY NETWORK

A distinct challenge for the task of forensic surveillance is the sheer volume of the data that needs to be reviewed for an operator to make a decision. For instance, the system deployed at MERL includes 6 video cameras. If an event of interest can be localized to within an hour, an operator might be facing the necessity to browse through 6 hours of video recording in order to review the evidence. Thus, the primary focus of our system is providing the operator with the ability to quickly assess the situation and find relevant evidence in a large volume of data.

The system that we built for managing the data collected with the sensor network includes the query engine with a gestural front end, as described in our earlier work.[3] Here we introduce the tracking extension that allows an operator track a single individual with some minimal amount of interaction with the system.

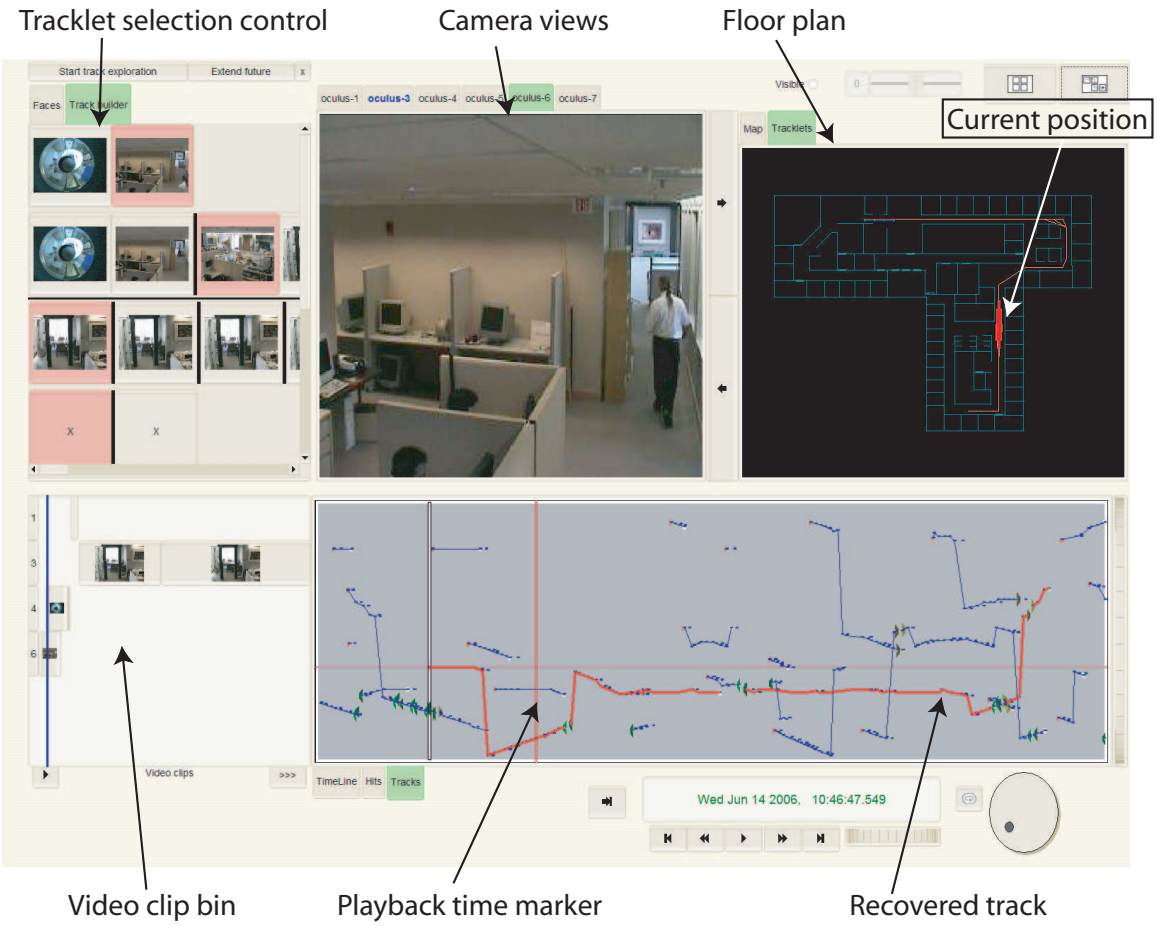

Figure 4. User Interface of the MERL Forensic surveillance system. The interface consists of five main panels, listed clock-wise - Floorplan, Timeline, Video clip bin, Tracklet selection control and Camera view, further described in the text.

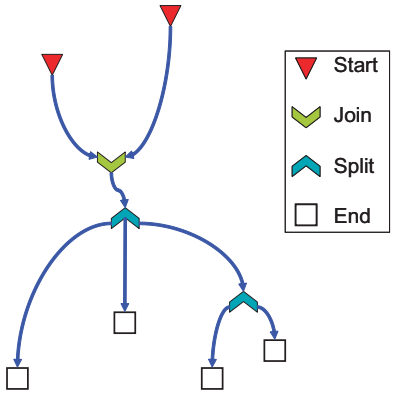

Figure 5. Tracklet graph representation of the track bundle. Each edge , called *Tracklet*, represents a contiguous sequence of sensor activations, while nodes - states of tracklet endpoints.

The screen shot of the tracking module is shown in Figure 4. The figure shows one of the camera views that is either selected by the operator, or automatically chosen by the simple camera scheduling algorithm. The scheduling algorithm is invoked during the playback of the clips form the video clip bin, shown in the bottom left of the interface. Upper right panel shows the floorplan with the currently selected track. The position of a person in the floorplan control is indicated by a "swell" in the track shown in the picture in red. Bottom gray panel is the "piano-roll" of the sensor activations, linked into *tracklets*, the elementary units from which *tracks* are built during the interaction with the operator. Finally, upper left panel is a visual representation of the *tracklet graph*.

4.1. Tracklets and Tracklet Graph

An idea that central to our philosophy of user assisted forensic tracking is the idea of *tracklets* and the corresponding representation of the tracklet set as a *tracklet graph*.

A *tracklet* is a set of sensor activations such that each sensor in the set is unambiguously reacheable from its immediate predecessor, according to the pre-determined geographic ordering. We will call the process of finding the immediate predecessor to a current sensor activation *linking*. A tracklet is an elementary building block of a track and has two end points. An end point is labeled with one of four labels:

1. *Track-Start*. The first sensor activation in the tracklet, such that no preceeding sensor activations can be linked to it within the predetermined time interval;
2. *Track-Join*. An end-point sensor activation in the tracklet such that there exist multiple preceeding tracklets that can be linked to it within the predetermined time interval*;
3. *Track-Split*. An end-point sensor activation in the tracklet such that there exist multiple successor tracklets that can be linked to it within the predetermined time interval†;
4. *Track-End*. The last sensor activation in the tracklet, such that it cannot be linked to any subsequent sensor activation within the predetermined time interval.

All tracklets form a set of graphs, each of which represents an inherent confusion of the system about individual tracks. A graph is a set of tracklets that can be joined according to the temporal and geographical restrictions which can be either imposed by the user or learned over time. A node in the tracklet graph is a label of a tracklet endpoint, while the edge represents the tracklet. Figure 5 shows an example of a track bundle

*a single valid predecessor tracklet may not exist as it would have already been linked into the current tracklet
†a single valid successor tracklet may not exist as it would have already been linked into the current tracklet

temporally arranged top-to-bottom. The graph has two starting tracklets, which subsequently merge into a single path. The merged tracklet then splits twice resulting in four end points.

The tracklet graph is the core representation of the sensor activation data that we use for the purposes of tracking. When the tracking information is stored in the database it is organized in tracklet graphs, which represent the limit of the system's tracking ability. When the need arises to extract a particular track these tracklet graphs are used to plan the efficient interaction with the operator and find the unambiguous interpretation of the scene.

4.2. Human-Guided Tracking

The task of human-guided tracking and forensic search that we attempt to solve with our system can be illustrated with a simple scenario:

A laptop was reported stolen from the office X during the lunch hour between 1:00pm and 2:00pm. There was no camera coverage available outside the office. The operator needs to find all people that passed by the office during the lunch hour, possibly identify them and collect evidence connecting the individual with the event.

In such a situation, the operator would want to identify all tracks that originated at the door of the office and to identify the suspect by collecting all available video evidence. It is to this end that we build our system.

4.2.1. General Principles of Forensic Tracking

Track-Start and Track-End labels are unambiguous beginnings and ends of complete tracks. However, automatic resolution of Track-Split and Track-Join label ambiguities is impossible in the space of sensor activations alone. The source of split and join labels is the perceptual blindness of the sensor network to any features other than presence of motion. In such situation, two people crossing paths in the hallway will cause the system to generate at least 4 tracklets, containing sensor activations for each person before and after the crossover point. Mapping the identity to these tracks and maintaining their continuity with absolute certainty is impossible. In the light of this we sidestep the mapping problem by using the following observations:

1. Operator does not need to disambiguate the entire scene, only the subgraph originating at the selected tracklet;
2. Resolving branch ambiguities can be simplified by considering video clips associated with each candidate track;

The first observation significantly reduces the amount of tracklets that need to be considered as possible candidates to be aggregated into the track. Since in our scenario an operator is tracking one person at a time, the system only needs to explain the behavior of that person, while effectively ignoring other occupants. That is, for the example of two people crossing paths, we assume one tracklet being selected before the cross-over, and therefore, only two tracklets need to be considered as a possible continuation, and not all four. This iterative, focused approach to tracking and track disambiguation allows us to reduce the complexity of the problem from potentially exponential to linear.

The second observation simply implies that when a split-join confusion occurs, the system can trace the tracklets to the nearest camera and display the corresponding video clips to the operator to make the decision about which tracklet is the plausible continuation for the aggregate track.

Even though one can imagine developing a tracking algorithm that estimates the dynamics of the motion of the objects under the network of sensors, any such algorithm will inevitably make mistakes. In security application the commitment to the results of the even slightly inaccurate tracking algorithm can be quite costly. Instead we chose to implement the tracking system following the "Human-In-The-Loop" metaphor,[12,13] using tracklet graphs as the underlying representation of the tracking data.

The main focus of the system is the efficient browsing of a large data set. To this end, we are concerned with decreasing the false negative rate, with false positive rate being a distant second. In order to achieve these goals we have adopted the mechanism of the track aggregation described below.

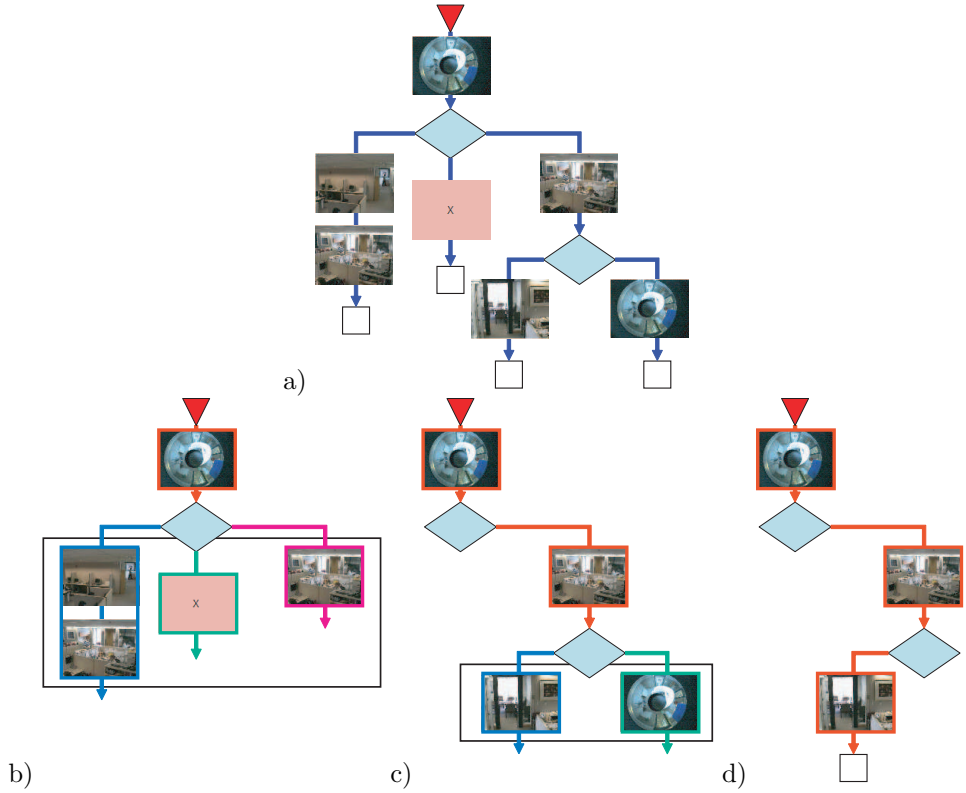

Figure 6. Human-guided track selection process using tracklet tree representation. a) Example of the selection subgraph which includes camera views available for each tracklet, as well as split/join locations where track splicing occurs (shown by blue diamond shapes). Tracklets are shown as edges of the graph passing through the camera views. b) First step of the interactive graph pruning process. One step-lookahead tracklets are presented to the operator. c) Second step of the graph pruning. d) Final track recovered.

4.2.2. Track aggregation

The process of human guided tracking in the forensic mode of our system begins with selecting a group of sensors where we expect the track to start. For instance, in our system, with sensor placed outside of people offices, the operator would select several sensors on the floor plan that can possibly be activated when the person leaves his office. An example of such a trigger condition is shown in the floorplan control in Figure 4. There the trigger event is set outside the office in the bottom part of the plan. By performing a very fast search in the database we can identify every instance a of a tracklet that originated in one of the chosen sensors.

At this point an operator needs to select a single track to explore. Upon selecting the first tracklet in the corresponding graph by simply clicking on the tracklet start in the timeline control, the tracklet is drawn on the floorplan up to the point where there is an end, a split or a join endpoint. If the endpoint is reached, then the track is declared complete. Otherwise, the process of track aggregation proceeds iteratively, using the tracklet graph to splice the candidate tracklets into a coherent track. In this process at each confusion point of the graph (split or join end point) the operator chooses the subgraph to traverse further. The process is illustrated in Figure 6. The thumbnails in the graph show that a video clip from the camera oriented towards the activated sensors is available. Blue diamond shape indicates that a confusion point is reached and there are possible conflicting tracklets following the confusion point. Links in the graph indicate that there exist a tracklet path.

Consider a subgraph shown in Figure 6a. The structure of this *selection graph* represents a set of paths through the tracklet set that is possible to traverse starting at the tracklet and the camera view shown at the top of the figure. Since the ambiguous points are known, at each such point the system can present the set of ambiguous tracklets to the operator for disambiguation. For instance, at the first step, the confusion point

Figure 7. State of the video clip bin after the complete track has been recovered. Each row of the bin represents a separate camera, while left-to-right arrangement reflects their temporal relationship. There are possible temporal gaps between the clips which are automatically removed for display purposes.

represents a 3-way split from the current node (Figure 6b). The first tracklet leads to two camera views in sequence. The second tracklet terminates without going through any camera, while the third one passes through a camera and leads to the subsequent 2-way split. Each of these tracklets is drawn on the floor plan and is colorcoded as shown in the figure.

Three groups of thumbnails are then added to the Tracklet selection control (see Figure 4). The color-coded thumbnail groups, as shown in Figure 6b represent a one-step lookahead on the full selection graph. The operator is asked to select one of the paths the he would like to explore further, thereby rejecting the other two. The resulting situation is shown in Figure 6c, where a 2-way split is further explored. If a mistake is made, the operator can use the selection tree control to roll back the selections. The process continues until the *end-track* label is encountered (see Figure 8).

Note that the tracklet selection graph in Figure 6a is related to the tracklet graph in Figure 5, but is not the same. In fact, the notation of the graph of Figure 6a represents a general selection mechanism, which can be used for traversal of the tracklet graph either forward in time (as illustrated) or backwards. In the former case the start and end markings of the selection graph in Figure 6a have the same meaning as those in the tracklet graph, while diamonds only represent splits. Track merges are irrelevant to the forward selection process, as they present no forward selection alternative. In contrast, if the selection graph is used for backward traversal, then start and end markings of the selection graph have the opposite meaning to those of the tracklet graph and diamonds only represent merges.

While the track is being built, the video clips related to each confirmed tracklet are collected in the Video clip bin (bottom left of Figure 4). The clip bin shows video clips ordered temporally left-to-right, while each row of the clips corresponds to an individual camera. Figure 7a shows the collection of video clips at the conclusion of the tracking task, corresponding to Figure 8). Note the occasional overlap between video clips from different cameras, which represent the situation when the person is observed by several cameras at once.

Figures 7b-f show selected frames from each of the clips in the clip bin after a complete track is built. They represent the entire set of the video evidence about the 15 minutes of person's movement during which he has been tracked. The rectangles outlining the person's position in each frame were added manually, for illustrative purposes.

The final assembled track, covering a 15 minute long interval, is shown in Figure 8 in red. The track begins outside the view of any of the cameras on the bottom left tip of the line (point 1). It then follows through 4 cameras with IDs 6, 4, 1 and 3 (Figure 7b-e respectively) to the upper right corner, where the person stops to talk to another occupant (point 2). Then the track proceeds all the way to the left, where the person disappears out of the sensor view for about 3 minutes (point 3). Finally, after a short period of hovering in the left-most end

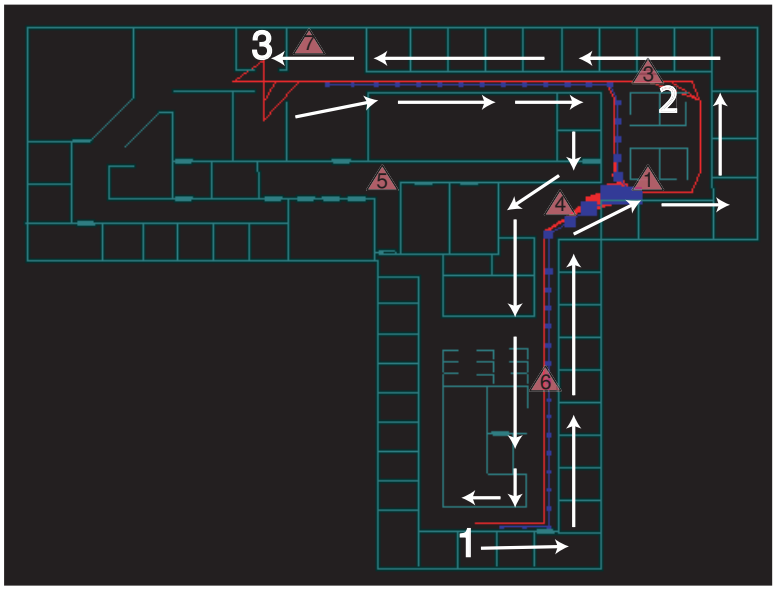

Figure 8. Final assembled track covering a 15 minute long time interval.

of the track the person retreats to his office via a different path, again passing through the view of the camera 4 (Figure 7f). Note that the camera 6 does not observe the person on its return trajectory, as at that time it turned away from the sensors in the path.

5. DISCUSSION AND CONCLUSIONS

The goal of the paper is to present a technique of efficient browsing and tracking of human in a large database of sensor activations and video feeds under the metaphor of Human-Guided Search.

In this work we presented our solution to the problem of tracking individuals inside an office space. As a primary tracking device we chose a distributed network of cheap sensors augmented with generally non-overlapping set of video cameras that move under an independent control. The video cameras only observe a small fraction of the entire space, which makes it difficult to use them alone for the purposes of tracking people. For example, tracking individuals that did not cross any of the camera views would be impossible.

In contrast, the sensor network provides a low-fidelity measure of all the activity in the space, with no gaps in space or time. This complete coverage makes tracking feasible and robust, despite the simplicity of the sensor modality. Furthermore this data simplicity is a distinct advantage in bandwidth and computational costs. The data is very compact, making the algorithms used to process it very fast and efficient.

We present our user interface and the concept of tracklet graphs. The tracklet graph models the inherent confusion in the tracking systems. This representation allows us to formulate the tracking problem as an iterative process of tracklet splicing in order to retrieve complete tracks of individuals along with all relevant video evidence collected along the track by any available camera.

We have seen that these algorithms, despite the simplicity of the sensor modality, can extract powerful descriptions of context from the data. Providing these bits of context to a human operator results in a much more efficient, reliable, and powerful system.

ACKNOWLEDGMENTS

We would like to thank Darren Leigh and Jonathan Westhues for the design and implementation of the sensor packs used in our sensor network.

REFERENCES

1. Mitsubishi Electric Security Products, DX-TL5000E. `http://www.bdt.co.nz/security/product.asp?item=610732`.
2. C. R. Wren, D. Minnen, and S. G. Rao, "Similarity-based analysis for large networks of ultra-low resolution sensors," *Pattern Recognition* **39**(10), pp. 1918–1931, 2006.
3. Y. Ivanov and C. R. Wren, "Toward spatial queries for spatial surveillance tasks," in *Pervasive Technology Applied: Real-World Experiences with RFID and Sensor Networks*, (Dublin, Ireland), 2006.
4. C. Stauffer and E. Grimson, "Learning patterns of activity using real-time tracking," *IEEE Transactions on Pattern Recognition and Machine Intelligence* **22**(8), pp. 747–757, 2000.
5. N. Johnson and D. Hogg, "Learning the distribution of object trajectories for event recognition," *Image and Vision Computing* **14**(8), 1996.
6. D. Minnen, I. Essa, and T. Starner, "Expectation grammars: Leveraging high-level expectations for activity recognition," in *Workshop on Event Mining, Event Detection, and Recognition in Video, held in Conjunction with Computer Vision and Pattern Recognition*, **2**, p. 626, IEEE, 2003.
7. R. Cutler and L. Davis, "Real-time periodic motion detection, analysis and applications," in *Conference on Computer and Pattern Recognition*, pp. 326–331, IEEE, (Fort Collins, USA), 1999.
8. T. B. Moeslund and E. Granum, "A survey of computer vision-based human motion capture," *Computer Vision and Image Understanding* **81**, pp. 231–268, 2001.
9. A. Wilson and A. Bobick, "Realtime online adaptive gesture recognition," in *Proceedings of the International Conference on Pattern Recognition*, pp. 111–6, (Barcelona, Spain), September 2000.
10. Y. A. Ivanov and B. M. Blumberg, "Solving weak transduction with em," *Robotic and Autonomous Systems* **39**(3), pp. 129–143, 2002.
11. D. H. Wilson and C. Atkeson, "Simultaneous tracking & activity recognition (star) using many anonymous, binary sensors," in *The Third International Conference on Pervasive Computing*, pp. 62–79, 2005.
12. D. Anderson, E. Anderson, N. Lesh, J. Marks, B. Mirtich, D. Ratajczak, and K. Ryall, "Human-guided simple search," in *Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence*, pp. 209–216, 2000.
13. L. Colgan, R. Spence, and P. R. Rankin, "The cockpit metaphor," *Behaviour and Information Technology* **14**(4), pp. 251–263, 1995.