Distributed Target Tracking using Self Localizing Smart Camera Networks

Babak Shirmohammadi
Department of Computer and Information Science
University of Pennsylvania
babaks@cis.upenn.edu

Camillo J. Taylor
Department of Computer and Information Science
University of Pennsylvania
cjtaylor@cis.upenn.edu

ABSTRACT

This paper describes a novel decentralized target tracking scheme for distributed smart cameras. This approach is built on top of a distributed localization protocol which allows the smart camera nodes to automatically identify neighboring sensors with overlapping fields of regard and establish a communication graph which reflects how the nodes will interact to fuse measurements in the network. The new protocol distributes the detection and tracking problems evenly throughout the network accounting for sensor handoffs in a seamless manner. The approach also distributes knowledge about the state of tracked objects amongst the nodes in the network. This information can then be harvested through distributed queries which allow network participants to subscribe to different kinds of events that they may be interested in. The proposed scheme has been used to track targets in real time using a collection of custom designed smart camera nodes. Results from these experiments are presented.

1. INTRODUCTION

The decreasing cost and increasing performance of embedded smart camera systems makes it attractive to consider applying them to a variety of surveillance and tracking applications. In the near future it will be possible to deploy small, unobtrusive smart cameras in the same way that one deploys lightbulbs, providing ubiquitous coverage of extended areas. We could imagine using such a system to track passengers at an airport from the time that they arrive at curbside check-in to the time that they board their flight. Similarly, we could use such a system to monitor the movements of elderly or infirm individuals in their homes in order to improve their quality of care.

In order to achieve our vision of a robust situational awareness percept derived from an ensemble of distributed cameras, we will need to address the problem of distributed sensing and tracking. More specifically, the challenge will be to reliably detect, localize and track targets as they move over an extended area of regard covered by multiple distributed smart cameras.

In order to field these kinds of systems we will need to develop approaches to detection and tracking which can be distributed over multiple sensors without requiring excessive amounts of communication. These systems must be scalable to allow for deployments that may involve thousands of cameras distributed over extended regions and must be robust to failure so that the overall system responds gracefully when individual sensors are added or removed asynchronously.

Most detection and tracking systems that have been developed or proposed fuse information from multiple sensors at a central point in the network which is responsible for establishing tracks and associating measurements from different views. As the number of sensors grows, increasing demands are placed on the communication system which must route information to these processing centers. Moreover failures in these processing centers can often render the entire network useless.

This paper describes a new approach to detection and tracking for smart camera networks which is fundamentally decentralized. This approach builds on previous work on self localization which allows the smart cameras to automatically detect and localize other camera nodes with overlapping fields of regard and to establish communication graphs which reflect how the nodes will interact to fuse measurements in the network. We have developed novel network protocols with limited communication requirements which allow the system to distribute the detection and tracking problem evenly through the network accounting for sensor handoffs in a seamless manner.

The approach also distributes knowledge about the state of tracked objects throughout the network. This information can then be harvested through distributed queries which allow network participants to subscribe to different kinds of events that they may be interested in. For example a process could request to be updated on all movements of a particular target or may want to be told about all targets that pass through a particular area of interest. These approaches can be used to develop simple but robust tracking systems that respect the constraints of a distributed deployment context.

The remainder of the paper is organized as follows: Section 2 briefly discusses related work on distributed tracking . Section 3 describes the implementation of the tracking protocol, Section 4 discusses some of the experimental results obtained with the method on our smart camera network and in simulation. Section 5 presents some of the conclusions drawn from the work.

ICDSC 2010 August 31 – September 4, 2010, Atlanta, GA, USA

2. RELATED WORK

Distributed tracking on smart camera systems has attracted a lot of recent attention and a number of groups have developed systems for this task [19, 20, 13, 21, 6, 9, 4]. For example Kayumbi, Anjum and Cavallaro [11] describe an effective scheme for localizing soccer players with a network of distributed cameras. Quinn et. al. [19] propose a scheme for calibrating a set of cameras in a room and using them to track targets. Their approach splits the tracking task between a collection of smart camera nodes and a higher level process which fuses the sightings from these cameras. In contrast, the goal of this paper is to develop protocols that can be employed on large networks covering extended areas.

More closely related to the approach described in this paper is the work of Medeiros, Park and Kak [15]. This paper describes a distributed approach to triangulating targets and distributing the tracking task over multiple nodes. This protocol involves electing a leader associated with every tracked object which is responsible for maintaining that track. Klausnet Tengg and Rinner [3] describe a distributed multilevel approach to fusing the measurements gleaned from a network of smart cameras. Their paper addresses the problem of automated aggregation of measurements through a hierarchy where different nodes have different capabilities and are given different responsibilities. The approach proposed in this paper is different from these methods since the cameras are all viewed as peers. There is no need for a leader election process nor is there any hierarchy. This simplifies the deployment procedure and the resulting protocol and results in a scheme which is simpler to implement and more resilient to failure since the state of the tracker is automatically replicated and distributed throughout the network.

Our approach builds on the work of Mikic et al.[16] and Focken et al.[8] who describe schemes for tracking objects in 3D by triangulating the sightings obtained from multiple distributed vantage points. Like these works we formulate the tracking problem as one of associating measurements from different cameras and establishing correspondences over time. Our work extends these approaches by leveraging previous work on self localization and by describing how the scheme can be scaled to hundreds or thousands of camera nodes.

Arth, Leistner and Bischof [2] describe a scheme for object tracking where tracks are associated between cameras in the network by extracting distinctive features and matching these features over widely distributed viewpoints. In contrast the approach described in this paper does not make use of distinctive appearance based features but instead assumes that the cameras are densely distributed in the scene so that targets can be handed off seamlessly.

Funiak Guestrin Paskin and Sukthankar [21] describe an interesting algorithm inspired by work on Simultaneous Localization and Mapping wherein they tackle the tracking and camera localization problems in a single unified framework. Their system is capable of both localizing the targets and the cameras with a single convergent procedure given a sufficient number of corresponding tracks. Our system leverages previous work on ad-hoc localization that allows the cameras to directly estimate their relative positions. This decomposition allows us to scale our approach more readily to larger networks since it allows us to avoid the problem of establishing correspondences between cameras in the absence of localization information and the complexities of uncertainty management in the SLAM approach.

When the images or results from multiple cameras can be processed at a central location, several sophisticated and effective algorithms have been proposed that provide state of the art results on the multi-camera tracking problem. See for example recent systems proposed by Liem and Gavrila [14], Eshel and Moses [5], Mittal and Davis [17], Khan and Shah [12], Arsic et al.[1] and Fleuret et al.[7]. In this work we consider what can be accomplished in the context where the tracking task must be carried out in real time by a distributed ensemble of embedded processors with limited communication bandwidth.

3. TECHNICAL APPROACH

The approach to target tracking proposed in this paper builds upon previous work on smart camera localization described in [22]. In this localization scheme each of the embedded camera systems is equipped with a controllable light source, typically an IR LED, and a wireless communication system. Each smart camera uses its signaling LED as a blinker to transmit a temporally coded sequence which serves as a unique identifier. The cameras detect other nodes in their field of view by analyzing image sequences to detect blinking pixels and, hence, are able to determine the relative bearing to other visible nodes. Figure 1 shows the simplest situation in which two nodes can see each other. Here we note that the accelerometer measurements provide another independent source of information about the orientation of the cameras with respect to the vertical axis. These measurements allow two smart cameras to determine their relative position and orientation up to a scale factor. When a collection of smart cameras is deployed in an environment, these visibility relationships induce a sparse graph among the cameras. These measurements can be used to localize the entire network up to scale. The scheme provides a fast, reliable method for automatically localizing large ensembles of smart camera systems that are deployed in an ad-hoc manner.

In addition to these visibility relationships, the smart camera nodes are also related by a sparse web of wireless communication links. These links also depend upon the layout of the sensors since signal strength falls off with increasing distance. Taken together, these visibility and communication links induce a natural notion of locality where each camera can discover, localize and communicate with other camera systems in its immediate vicinity. This localization capability provides a basis on which each camera can correlate its own image measurements with those obtained from other neighboring vantage points and, hence, localize targets moving through its area of regard.

3.1 Target Tracking

Unlike other sensor modalities, the measurements obtained from camera systems are most useful when they are combined together as depicted in Figure 2. Here we see a typical situation where a target moving through the scene is localized in space by combining bearing measurements obtained from a variety of vantage points. Each bearing measurement is referred to as a sighting and for each sighting the camera can determine the range to the associated target by looking for confirming evidence from at least two other cameras. This notion of collaborative tracking is commonly employed

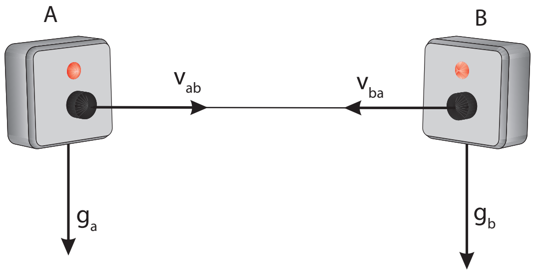

Figure 1: Two smart cameras equipped with accelerometers that can see each other can determine their relative position and orientation up to a scale from the available measurements. Larger networks can be localized by leveraging this relative localization capability.

in a number of vision based tracking systems [14, 7, 12, 17, 8, 16] and is most easily done when the bearing measurements can be relayed to a central location for processing.

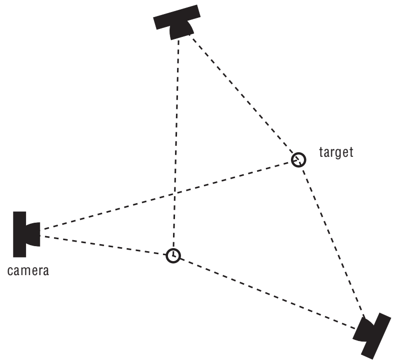

Figure 2: The bearing measurements obtained from two or more smart camera systems can be fused via triangulation to determine the position of the targets in three dimensions. When three or more smart camera systems are viewing the same area of space redundant measurements are available which can be used to help eliminate false tracks.

In this work we prefer a distributed approach where each camera localizes targets on its own by communicating with its neighbors. This is accomplished by having each node communicate its sightings to all of the other nodes in its immediate neighborhood. Once this has been done, each camera independently considers every pair of sighting measurements that it learns about, for each of the resulting candidate points it looks for confirming sighting measurements in other views. The end result of this procedure is a set of candidate points in space which we term **targets**.

In addition to the set of targets derived from the sighting measurements, each camera also maintains a set of active **tracks** corresponding to trajectories of targets over time. Each of these track structures contains a number of fields which are described in Table 1.

Associated with each track is a state vector which encodes

(x, y, v_x, v_y)	State vector encoding the position and velocity of the object.
C	Covariance matrix associated with the state estimate
track id	Globally unique identifier associated with the track. The smart camera that creates this track forms this number by concatenating its own unique identifier with a local counter value to yield a globally unique identifier.
first timestamp	Time when the track was first created
last timestamp	Time when the track was last updated with a sighting

Table 1: Fields associated with each track data structure

the position and velocity of the associated target along with the covariance matrices that are required to implement a Kalman filter tracker. Each track is tagged with a globally unique identifier which persists as the object moves through the entire scene. The track structure also contains timestamps indicating the first time that the tracked object was detected and the last time that the track was updated.

On each cycle every smart camera system must solve a data association problem resolving the relationship between the current tracks and the current targets. We can model the situation in terms of a bipartite graph as shown in Figure 3 . Here the nodes on the left depict the current tracks while the nodes on the right depict the current targets. For each track we determine the most probable target based on the Kalman filter estimate and covariance and link the nodes as shown. Note that at this stage there may be more than one track associated with each target. Each target then chooses its best match among the list of possible tracks by considering the relative age of the tracks and choosing the oldest one. The measurements associated with this target are then used to update the Kalman filter associated with the winning track. Tracks that are not updated are propagated forward allowing for short periods of occlusion or tracker failure. Tracks that are starved of updates are eventually elided from the list. In this manner, short ephemeral tracks are removed from the system in favor of longer lasting records. In this scheme we assume that the clocks on the smart camera nodes are roughly synchronized so that timestamps can be compared without issue. This scheme is similar to the best-hypothesis tracking scheme described in [8].

Detected targets that are not explained by any of the existing tracks are used to create new tracks. When a smart camera creates such a track it concatenates its own unique smart camera identifier with a local counter value to form a globally unique identifier. This global identifier is then used in all subsequent communications and effectively travels with the object as it moves through the network.

Once the camera has resolved the relationship between tracks and targets and updated the list of tracks, it sends its current list of active tracks to its neighbors receiving in turn a list of all the targets that they are tracking. In this manner, information about tracks and target identities is propagated through the network allowing for seamless hand-

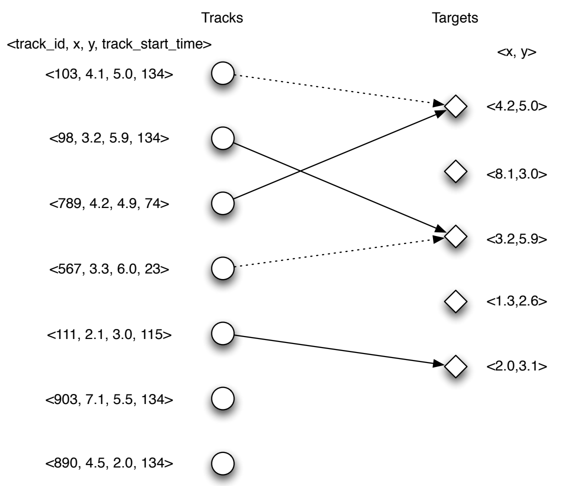

Figure 3: With every new image, each smart camera node must associate the detected targets with the tracks that it currently maintains. For each of the current tracks the system finds the best matching target - if any. The system then selects between multiple associations by favoring tracks with longer histories. In this figure the ultimate matches are indicated by solid lines while the dashed lines indicate possible matches that are rejected. This scheme causes the overall system to maintain the identities of the tracked objects as they move through the system. Unmatched targets become new tracks while unmatched tracks are eventually elided.

off as targets move throughout an extended scene. Note that ephemeral tracks may be introduced from time to time due to tracker failures or glitches but these are typically corrected eventually since the system is biased to prefer older labels for tracks wherever possible. We also expect occasional miss-associations in the triangulation phase. These cases typically produce outlier targets which do not win data association competitions and are thus starved for updates. As stated earlier, tracks which do not receive a sufficient number of updates per second are elided. (In the current implementation a track must receive at least 2 updates per second).

This protocol allows the network of distributed, loosely coupled smart camera systems to achieve consensus on target identities. By preserving target identifiers, the entire system is able to track targets over extended areas without requiring a central coordinating authority. Moreover, since the protocol only requires communication among near neighbors it can be scaled to networks of arbitrary size.

The entire procedure carried out by the smart camera is outlined in Algorithm 1 in pseudo-code.

In summary, on every iteration of the tracking algorithm each camera sends to its neighbors a list of its current bearing measurements (step 3 of the algorithm). It also sends out a list of its current tracks (step 8). Each track structure contains the fields described earlier. Since these messages contain relatively little information and each camera only communicates with its near neighbors the method makes efficient use of the available communication bandwidth which is often quite limited.

Algorithm 1 Distributed Tracking Protocol

```
1: loop
2:    Process current image and extract sightings
3:    Gather sighting measurements from neighboring cameras
4:    Triangulate sightings to obtain positions of current targets
5:    Match current targets against the list of current tracks
      and update the tracks that are matched.
6:    Targets which are not matched with any existing
      tracks are used to form new tracks.
7:    Prune tracks that have not been updated recently.
8:    Gather current tracks from neighboring cameras removing duplicates as needed
9: end loop
```

3.2 Exfiltrating Information

An interesting feature of the proposed protocol is that the information about the targets in the scene is distributed among the smart cameras in the network. In fact information about the trajectory of a particular target is distributed among the smart camera nodes that viewed the target at various portions of its trajectory. These trajectory are linked by a common target id which can be used to correlate the information across the network.

In order to harvest information from the network we propose a subscription model where a user can inject a request into the network indicating her interest in particular kinds of tracking events. This request would be broadcast periodically through the prevailing communication network to the individual camera nodes which would respond by sending back events that matched the criterion of the query.

For example a user may indicate that she is interested in all tracks passing through a particular region in the scene or all tracks that persist for more than a certain amount of time. Alternatively she may indicate interest in a particular set of target ids which could then be tracked over time with the information relayed back to the subscriber.

This scheme would decouple the service providers, the individual smart camera nodes, from the service subscribers. The subscribers would not request information from particular nodes which could fail or be removed at any time but would rather phrase their request in terms of queries which would be broadcast to all of the nodes that may be able to provide them with the desired information. This would mean that individual smart camera nodes would be able to change their mode of operation without disabling client applications. It also implies that the network could service multiple applications for multiple subscribers concurrently.

4. EXPERIMENTAL RESULTS

4.1 Smart Camera Results

In order to test the proposed tracking scheme we designed, built and deployed a set of customized smart camera nodes one of which is shown in Figure 4. Each smart camera system is powered by a dual core 600 MHz Blackfin processor from Analog Devices. This Digital Signal Processor was designed to support high performance image processing operations in low power devices such as cameras and cell phones. The smart camera board can be interfaced to a range of Aptina CMOS imagers, in our experiments each camera was outfitted with a XVGA resolution imager (720x480) and a fisheye lens which affords a field of view of approximately 180 degrees. The system is also outfitted with a Zigbee wireless communication module, an Ethernet controller, a three axis accelerometer and an 850 nm high intensity infrared signaling light. The unit consumes less than 3 watts of power in operation and can be powered for 6 hours with a 6 ounce Lithium Ion battery pack.

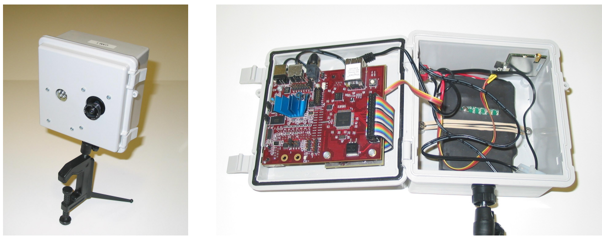

Figure 4: Argus Smart Camera Node used in our experiments.

In our experiments a set of 8 cameras were deployed in an ad-hoc manner to cover the entire first floor of our office building, an area approximately 14 meters on side shown in Figure 5. The cameras were automatically localized as described in [22] and then used to track targets moving through the area of regard in real time.

The first stage in the target tracking procedure is an adaptive background subtraction phase which determines which aspects of the image have changed significantly over time. A connected component phase is applied to the resulting binary image to find the most significant moving targets in the scene. The result of this analysis is a set of bearing vectors emanating from each of the cameras into the scene. Importantly, all of the real-time image processing occurs on the smart camera nodes themselves. Only the final sighting vectors associated with the targets are relayed over the Zigbee wireless network for further processing. This approach allows us to distribute the most computationally intensive aspects of the tracking procedure onto the smart camera nodes and avoid having to relay video imagery over the limited wireless bandwidth. Currently, each smart camera system extracts sighting measurements from the images at a rate of 15 frames per second.

Figure 5: Snapshots of the first floor area showing some of the deployed cameras

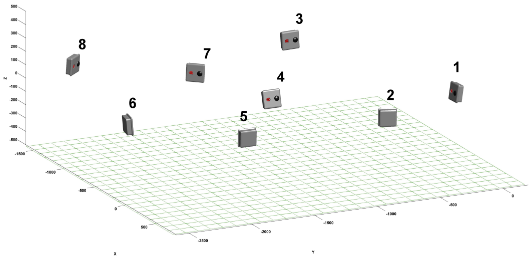

Figure 6: The smart cameras automatically determine their positions and orientations with respect to each other in 3D in a matter of seconds.

Figures 7 and 8 shows the results of two tracking experiments. In the first experiment the camera network was used to track a single person who walked into and around the first floor area and eventually exited through the same point he entered. The second experiment shows the system tracking two people who start off walking together and then split off onto two separate trajectories. (Videos which show the entire tracking sequence are provided in the supplementary material). In both of these experiments the targets were correctly tracked and associated throughout the sequence in real time through multiple camera handoffs since the structure of the scene ensured that no one smart camera had the targets in view throughout.

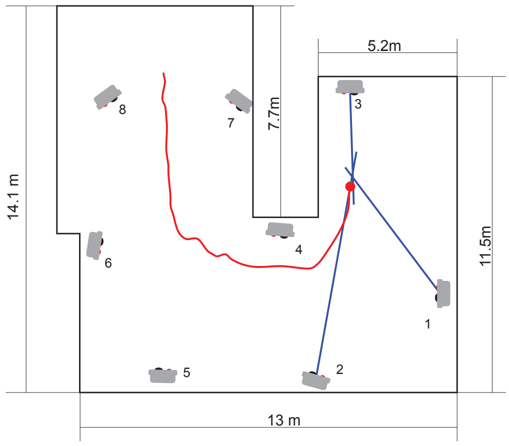

Figure 7: Snapshot of a real time tracking experiment showing the current position and past trajectory of a target moving through the scene. The lines emanating from the cameras represent sighting measurements which are triangulated to determine target location.

4.2 Simulation Results

In addition to the actual implementation on our smart camera network, a series of simulation experiments was carried out to investigate how the proposed scheme would per-

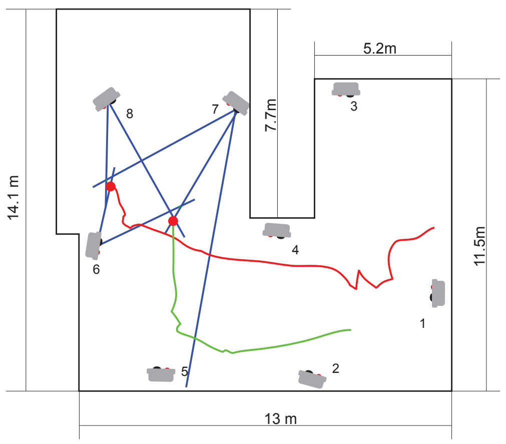

Figure 8: Snapshot of a real time tracking experiment showing the current position and past trajectory of two targets moving through the scene. Note the spurious sighting measurement, caused by a reflection in the window, which is not confirmed by the other cameras and, hence, discarded.

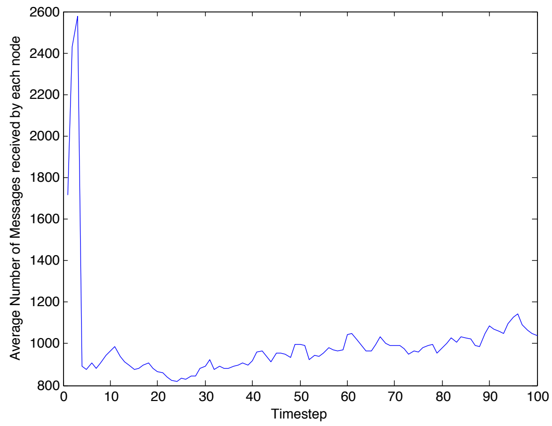

Figure 9: This graph indicates the average number of messages received by each of the nodes over the course of the simulation

form on networks that were considerably larger than the ones we could construct with our available hardware. Figure 10 shows an example of an indoor environment reminiscent of an airport. This particular scene is monitored by a collection of 168 cameras mounted along the walls. The cameras could only see targets within a fixed distance of their positions. Using the proposed protocol, the system was able to concurrently track a collection of 100 simulated targets over 100 timesteps. In order to characterize the communication requirements of the protocol the average number of messages received by each node was recorded on every timestep and the results are plotted on the graph in Figure 9. This plot includes both types of messages exchanged over the network, sighting measurements and track information. After an initial transient where the nodes communicate a number of measures to achieve consensus on target identifiers, the system settles into a steady state. The communication load here is evenly distributed throughout the network reflecting the distribution of the targets.

Figure 10 shows the trajectories recovered for a few of the tracked targets. Note that the targets move throughout the environment between many cameras but the tracking system correctly maintains their identities.

In order to investigate how the protocol performed in the presence of failure the simulation experiment was repeated with the additional wrinkle that on each timestep each of the simulated smart cameras had a ten percent chance of failure. A camera that fails produces no measurements in the network. Even in this situation the tracker was able to track 87 percent of the targets correctly through all 100 of the timesteps in the simulation. The remaining thirteen continue to be tracked but their identities are changed from the first to the last timestep indicating a tracker reacquisition. The resilience of the system to individual camera failure is a product of the fact that the tracker state is naturally distributed among a number of nodes so the failure of any single node is not catastrophic.

5. CONCLUSION

This paper describes an approach to using a network of distributed self-localizing smart cameras to localize and track moving obstacles in the scene. Our approach takes the view that smart cameras are currently small enough and cheap enough that one can consider deploying them fairly densely much as one installs lightbulbs. The challenge then is one of coordinating their activities so as to extract useful information from the ensemble subject to the prevailing computational and communication limitations.

In this approach each of the cameras independently analyzes its video imagery to find moving targets in its field of view, the results of this analysis are fused in the network to triangulate the location of the objects of interest in space. This approach devolves all of the low level image processing to the smart cameras and allows the nodes to use the limited wireless bandwidth more efficiently since they need only share sighting measurements and track data with their near neighbors. Using this approach we have been able to demonstrate real time tracking of targets over an extended area using a collection of embedded smart cameras, deployed in an *ad-hoc* manner and connected by a wireless communication network.

Currently the memory architecture and limited computational power of our smart camera nodes constrains what can be implemented in real time on our network. More powerful processors would allow for more sophisticated target detection schemes such as face detection or pedestrian recognition which would cut down on false sightings. Nonetheless, the current implementation demonstrates that the proposed fusion scheme is resilient to spurious sightings from individual cameras because of the cross validation.

Importantly the proposed scheme is completely distributed, all of the nodes behave as peers and the tracking computations and tracking results are distributed throughout the network. Nonetheless, the protocol allows the networked cameras to achieve distributed consensus on the target identifiers which allows the system to seamlessly track targets as they move throughout the scene. The system is robust to isolated failures of individual nodes since multiple cameras

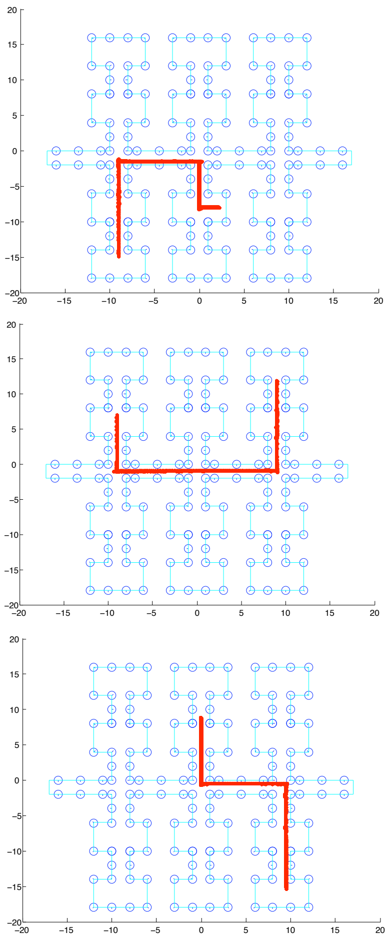

Figure 10: This simulation experiment modeled the layout of an airport. The system successfully tracked 100 targets using 168 smart camera nodes. The figure shows trajectories of individual targets successfully tracked throughout the environment. The small circles denote camera positions, the light lines indicate walls.

typically cover any area and it recovers gracefully from momentary tracker failures.

The approach leverages the fact that the nodes can recover the relative position and orientation of their neighbors automatically. This makes it feasible to consider deploying large collections of smart camera nodes in an *ad-hoc* manner since one need not manually survey their relative locations. Furthermore, it allows the smart cameras to rapidly and reliably determine the nodes with which it must collaborate in the tracking application. This drastically reduces the cost and complexity of fielding multi-camera surveillance systems and allows them to be applied to a wider range of applications.

6. REFERENCES

[1] D. Arsic, E. Hristov, N. Lehment, B. Hornler, B. Schuller, and G. Rigoll. Applying multi layer homography for multi camera person tracking. In *Distributed Smart Cameras, 2008. ICDSC 2008. Second ACM/IEEE International Conference on*, pages 1–9, Sept. 2008.

[2] B. H. Arth C., Leistner C. Object reacquisition and tracking in large-scale smart camera networks. In *Proc. First ACM/IEEE International Conference on Distributed Smart Cameras (ICDSC '07)*, pages 156–163, Vienna, Austria, Sept 2007.

[3] K. A. T. A. R. B. Distributed multilevel data fusion for networked embedded systems. In *Selected Topics in Signal Processing, IEEE Journal of Publication Date: Aug. 2008*, Aug 2008.

[4] A. Branzan-Albu, D. Laurendeau, S. Comtois, D. Ouellet, P. Hebert, A. Zaccarin, M. Parizeau, R. Bergevin, X. Maldague, R. Drouin, S. Drouin, N. Martel-Brisson, F. Jean, H. Torresan, L. Gagnon, and F. Laliberte. Monnet: Monitoring pedestrians with a network of loosely-coupled cameras. In *International Conference on Pattern Recognition (ICPR 2006)*, Hong Kong, China, August 20-24 2006.

[5] R. Eshel and Y. Moses. Homography based multiple camera detection and tracking of people in a dense crowd. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8, June 2008.

[6] S. Fleck, F. Busch, P. Biber, and W. Strasser. 3d surveillance a distributed network of smart cameras for real-time tracking and its visualization in 3d. In *Embdded Computer Vision 06*, page 118, 2006.

[7] F. Fleuret, J. Berclaz, R. Lengagne, and P. Fua. Multicamera people tracking with a probabilistic occupancy map. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2):267–282, 2008.

[8] D. Focken and R. Stiefelhagen. Towards vision-based 3-d people tracking in a smart room. In *Proceedings of the Fourth IEEE International Conference on Multimodal Interfaces (ICMI 02)*, 2002.

[9] N. Gehrig and P. L. Dragotti. Distributed sampling and compression of scenes with finite rate of innovation in camera sensor networks. In *in Proceedings of Data Communication Conference (DCC), Snowbird, Utah, USA*, pages 83–92, 2006.

[10] S. Hengstler and H. Aghajan. A smart camera mote architecture for distributed intelligent surveillance. In *Workshop on Distributed Smart Cameras (DSC 06)*, October 2006.

[11] G. Kayumbi, N. Anjum, and A. Cavallaro. Global trajectory reconstruction from distributed visual sensors. In *International Conference on Distributed Smart Cameras 08*, pages 1–8, 2008.

[12] S. M. Khan and M. Shah. A multiview approach to tracking people in crowded scenes using a planar homography constraint. In *In European Conference on Computer Vision*, 2006.

[13] T. Ko, Z. M. Charbiwala, S. Ahmadian, M. Rahimi, M. B. Srivastava, S. Soatto, and D. Estrin. Exploring tradeoffs in accuracy, energy and latency of scale invariant feature transform in wireless camera networks. In *Proceedings of ACM/IEEE International Conference on Distributed Smart Cameras*, pages 313–320, 2007.

[14] D. M. G. Martijn Liem. Multi-person tracking with

overlapping cameras in complex, dynamic environments. In *British Machine Vision Conference*, 2009.

[15] H. Medeiros, J. Park, and A. Kak. Distributed object tracking using a cluster-based kalman filter in wireless camera networks. *IEEE Journal of Selected Topics in Signal Processing*, 2(4):448–463, August 2008.

[16] I. Mikic, S. Santini, and R. Jain. Tracking objects in 3d using multiple camera views. In *Asian Conference on Computer Vision (ACCV)*, January 2000.

[17] A. Mittal and L. S. Davis. M2tracker: A multi-view approach to segmenting and tracking people in a cluttered scene. *Int. J. Comput. Vision*, 51(3):189–203, 2003.

[18] M. Quaritsch, M. Kreuzthaler, B. Rinner, and B. Strobl. Decentralized object tracking in a network of embedded smart cameras. In *Workshop on Distributed Smart Cameras (DSC 06)*, October 2006.

[19] M. Quinn, R. Mudumbai, T. Kuo, Z. Ni, C. D. Leo, and B. S. Manjunath. Visnet: A distributed vision testbed. In *ACM/IEEE International Conference on Distributed Smart Cameras, 2008*, pages 364–371, Sep 2008.

[20] B. Song, C. Soto, A. Roy Chowdhury, and J. Farrell. Decentralized camera network control using game theory. In *International Conference on Distributed Smart Cameras*, pages 1–8, 2008.

[21] G. R. S. Stanislav Funiak, Carlos Ernesto and M. Paskin. Distributed localization of networked cameras. In *Fifth International Conference on Information Processing in Sensor Networks (IPSN'06)*, pages 34 – 42, April 2006.

[22] C. J. Taylor and B. Shirmohammadi. Self localizing smart camera networks and their applications to 3d modeling. In *ACM SenSys/First Workshop on Distributed Smart Cameras (DSC 06)*, October 2006.