We thank reviewers for their constructive comments.

The reviewers expressed the following main concerns, which we have addressed accordingly in the revised report:

1. The discrepancy between our definition of a group leader, identified by the social network analysis algorithm, and the “real” leader of gang or criminal enterprise.

   As pointed out by the reviewers, the real leader often “will protect his or her exposure to detection with the use of runners”, “be embedded in the group”, or “may not even be in scene”. However, our notion of a group leader is borrowed from the social network research community where a node is identified as a highly influential node if it connects with many other nodes in the graph. We have clarified this in the report, in Section 2.2, on page 8.

2. Concerns about the performance of our system in real world testing.

   We discussed field testing and related performance issues such as false positives and “the extent to which this work can be used for higher crowd density” in Section 9.

We have also revised the following part of the report on presentation:

3. We have moved the technical details of the PTZ camera control to a standalone section in the Appendix.

4. We have moved the technical details of the social network analysis and computation to a standalone section in the Appendix.

5. General polishing on presentation and exposition.

GE
Global Research

Automated Detection and Prevention of Disorderly and Criminal Activities

Nils Krahnstoever (PI)

Final Report

Sensor Surveillance Program
Grant Number 2007-RG-CX-K015
Reporting period: January 2008 to October 2009

Submitted to
U.S. Department of Justice
Office of Justice Programs
National Institute of Justice
810 Seventh Street N.W.
Washington, DC 20531

Submitted by
GE Global Research
One Research Circle
Niskayuna, NY 12309

Technical Point of Contact
Xiaoming Liu
Principal Investigator
Phone: (518) 387-5346
Fax: (518) 387-5975
liux@ge.com

Administrative Point of Contact
Sarah M. Stotz
Business Development Manager
Phone: (518) 387-5444
Fax: (518) 387-5449
stotz@research.ge.com

August 15, 2011

imagination at work

Table of Contents

1 Abstract

This document is the final report for the NIJ research program "Automated Detection and Prevention of Disorderly and Criminal Activities". The goal of this program is to develop methods for automatically detecting and preventing criminal and disorderly activities using an intelligent video system. A particular emphasis of this program is to develop methods that can operate in crowded environments such as prisons, public parks and schools where a large number of people can be present and interact with each other. In addition, the developed technology is going beyond simple motion-based behavior features toward estimating meaningful *social* relationships between people and groups and use of this information for semantically high-level behavior and scenario recognition.

Some of the accomplishments of this program are: (i) a collection of crowd parameter estimation algorithms was developed that allows the system to estimate information such as crowd and group size, crowd density, and group velocity from video; (ii) motion pattern analysis algorithms were developed for detecting low-level group and crowd events such as GROUP FOLLOWING, GROUP CHASING, FAST GROUP MOVEMENTS, and GROUP FORMATION and GROUP DISPERSION; (iii) higher-level behavior recognition algorithms have been developed for detecting and predicting events such as FIGHTING and AGITATION; (iv) an algorithm for automatically controlling a network of PTZ cameras has been developed that enables face detection and face recognition of non-cooperative individuals from a distance; (v) a novel framework for estimating social network structures of groups from video has been developed that enables the system to determine the number of social groups and the leadership structure in small communities automatically; (vi) the system was tested live during the 2009 Mock Prison Riot sponsored by the NIJ as well as evaluated against a large amount of highly-relevant video data that was collected during the same event. The deployed system was demonstrated to law enforcement and correctional staff and received high praise for it's performance and innovation.

Overall this program has led to the development of a wide range of intelligent video capabilities that are highly relevant to law enforcement and corrections. The developed technology can help

All Mock Prison Riot data was *enacted* by law enforcement and corrections practitioners.

law enforcement detect many different types of activities and alert operators in many cases about the *onset* of an event – enabling early detection and possibly prevention of critical events. The system will also allow law enforcement gain insight into the ways that people behave and interact as well as into the social structure behind their interactions. Knowledge about social relationships enables the prediction and detection of challenging group events, such as gang activity and in the future the presence or formation of open-air drug markets.

All Mock Prison Riot data was *enacted* by law enforcement and corrections practitioners.

2 Executive Summary

This progress report covers the time period from January 2008 through October 2009. In the following, an executive summary of the extended report is provided.

2.1 Data Collection

During this program, video data was drawn from three sources. At the onset of the program, an initial data collection was performed with a group of volunteers recruited from the general population of GE employees and contractors. This first dataset focused on motion patterns that occur in crowded environments such as group dispersion and formation, chasing, following and others. In addition, data from existing public video repositories was utilized for this study. Finally, toward the end of the program and additional data collection was performed at the 2009 Mock Prison Riot, sponsored by the NIJ. During this very successful event, about 23.7 hours of high-quality video material was captured from four cameras, amounting to more than $120GB$ of video material. See Figure 1 for an overview of the datasets.

All data collections performed under this program have been evaluated and approved by an IRB board.

2.2 Crime Detection and Prevention

The goal of this program is to develop methods for automatically detecting and preventing criminal activities. The developed technology is intended to go beyond simple motion based behavior features toward a semantically high-level of understanding of interesting activities by estimating meaningful *social* relationships between people and groups interacting in crowded environments. Two of the main technical challenges in this program were:

- Detecting events as well as motion and behavior patterns from tracks of people in crowded environments. Common group and crowd motion patterns needed to be recognized and parameters such as crowd size, crowd velocity, agitation level, and events such as group

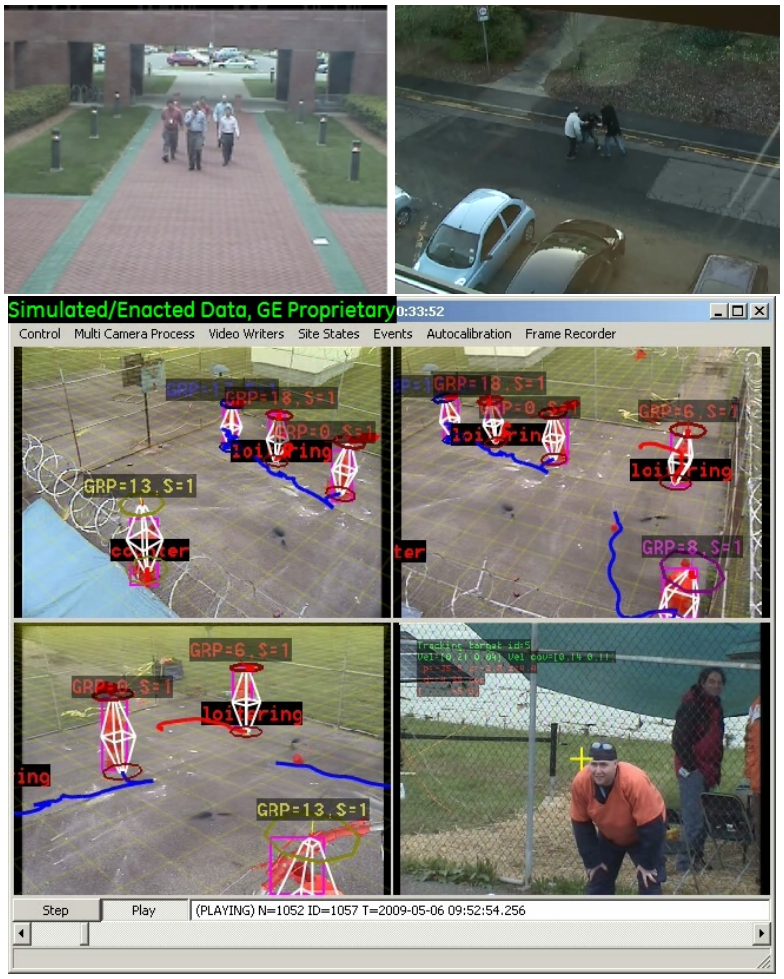


Figure 1: Datasets. Three examples of data utilized as part of this program. Top Left: data collected at the GE Global Research Center (GRC). Top Right: the public "BEHAVE" dataset. Bottom: data collected at the 2009 NIJ Mock Prison Riot.

All Mock Prison Riot data was *enacted* by law enforcement and corrections practitioners.

formation and dispersion needed to be detected. The detection and recognition needs to occur on noisy data obtained by a video tracking system.

- Establishing identity records of individuals based on their facial images in non-cooperative environments. Once identity records are established, associations (interactions) between subjects can be recorded and, over time, association graphs be build that represent the social connections between individuals, i.e., the *social network structure*.

Next we will provide an overview over the various motion and behavior pattern recognition components as well as the facial capture and social network estimation algorithms developed under this program.

Counting and Crowd Detection: Based on the detection response from the GRC tracking system we developed a set of tools that report the size (number of people) of a crowd and the crowd density. These tools can notify an operator when a crowd gathers at a certain location or if the density of a crowd has reached a level where conditions (due to egress concerns or due to the possibility of violence) are no longer safe. Figure 2 shows an example where the number of people in front of an airport exit has been estimated across time. The same approach could be used in many other places to monitor crowd density to prevent the outbreak of aggression or riots.

Group-Level Events: Under this program we developed a framework that continuously performs a group-level analysis of all tracking data. More specifically, we first combine (cluster) the continuously changing tracks of individual people into groups of people. We then analyze the (possibly continuously changing) groupings of individuals for a variety of patterns. One such pattern is for example FOLLOWING, where an individuals or group is followed by another group. Similarly, events such as CHASING (which we define as *fast* following) have been developed. See Figure 3 for an example of a chasing event. These motion pattern recognition algorithms are useful for the early detection of assaults. For example in a remote area of public park, the system could detect when a person is being chased by one or more individuals. In addition, we developed detection algorithms for group-level motion patterns representing GROUP FORMATION and GROUP DISPERSION. These low-level pattern detection algorithms can also be used in conjunction with other algorithms to detect more complicated behaviors such as the covert approach by a prison

All Mock Prison Riot data was *enacted* by law enforcement and corrections practitioners.

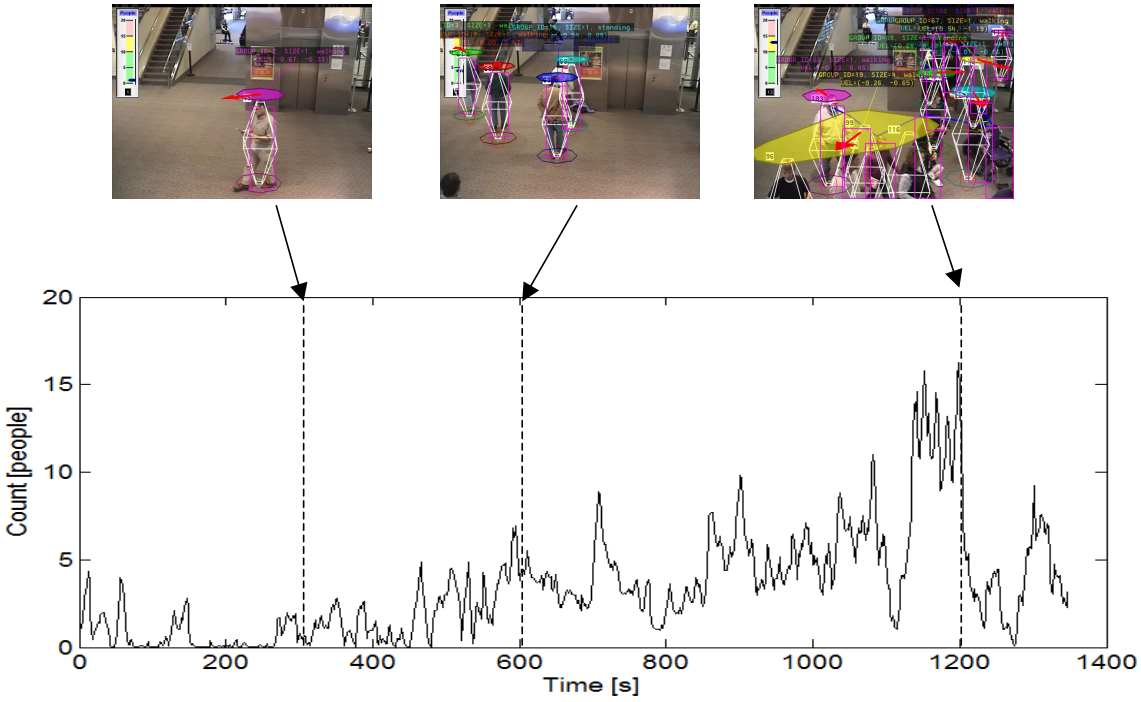


Figure 2: People Counting in Crowded Conditions. Crowd detection and counting applied to an Airport environment.

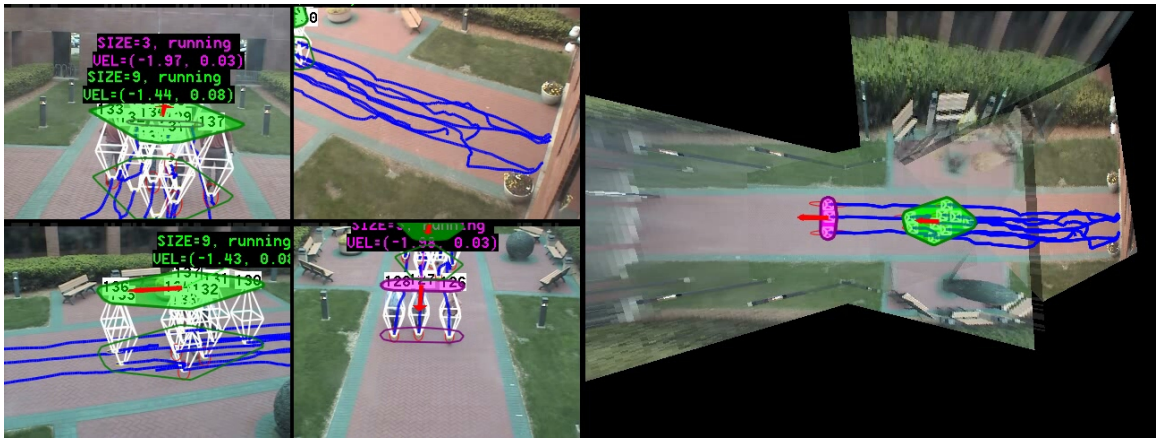


Figure 3: Crowd Motion Pattern - CHASING**.** Visualization of the grouping and crowd analysis stage. A larger group chases a small group. Both are running. The colored regions visualize the clustering result. The right side of the display shows a virtual top-down visualization of the surveillance site. Red arrows indicate the speed and direction of travel.

All Mock Prison Riot data was *enacted* by law enforcement and corrections practitioners.

gang toward an unsuspecting correctional officer or inmate. One such additional detection algorithm was developed in response to the experience gained at the Mock Prison Riot, where fights between gangs often began by one gang (the aggressor gang) approaching another (the defending gang) by extending its "flanks". We hence termed this pattern "flanking maneuver" and developed a detection algorithm that can detect it and hence predict and prevent the onset of violence. Figure 4 shows an example where the flanking maneuver between two gangs was detected shortly before a fight (simulated and enacted) between these gangs began. See Section 8.6.5 for details.

Figure 4: Flanking Detection. Example where the system correctly detected a FLANKING event and hence provided an alert to the operator *before* the subsequent fight ensued.

Agitation and Aggression Detection: We have developed an approach to agitation and aggression detection from video. The approach is based on tracking a set of salient points in the *foreground* of the scene and to perform aggression/non-aggression classification based on features extracted from these tracks. We utilize the FAST feature detector developed by Tom Drummond [1] for low-level point detection. The approach uses a heuristic to determine high-textured points but is extremely fast compared to other approaches. To obtain trajectories for the detected points, we developed a data association-based point feature tracker. After the tracking step, every trajectory is analyzed with regards to a range of motion attributes and the attributes are accumulated in local "decision blocks" (small regions in the image of size 16×16). The per-block features are then classified according to a learned agitation model. We utilize a Support Vector Machine trained on a small

All Mock Prison Riot data was *enacted* by law enforcement and corrections practitioners.

number of example sequences to obtain an optimal classification. Figure 5 shows a set of examples where aggression was detected in scenarios that were captured during the 2009 NIJ Mock Prison Riot.

The aggression/agitation detection process operates in real-time. Quantitative experiments based on Mock Prison Riot data have shown that the detection rate is around $DR = 0.85$ at a false alarm rate of $F = 0.50$ false alarms per true alarm. Missed detections occur in situations where the aggression is a quick stabbing and not much struggle/fighting occurs before and after the event. False alarms are caused by issues such as basketball play (which in some sense is not an incorrect indication of excessive aggression), and inmates hitting the fence, which causes the entire camera equipment to shake vigorously.

Social Network Estimation: One goal of this program was to extend the reasoning of the system into areas where the identity of individuals is taken into account. The motivation for this was that certain semantically complex circumstances can only be understood and detected if interactions between individuals are taken into consideration. We have hence developed a framework for visually estimating the social network structure that represents the interactions between individuals in a closed community. For this we had to solve two major challenges: (i) in order to be able to reliably maintain the identity of tracked individuals we had to perform face detection and face recognition at a distance. We developed a system that allows us to control a network of pan-tilt-zoom cameras to obtain imagery of subjects that is of sufficient quality and resolution to perform face recognition. Furthermore (ii) we developed an approach to monitoring the interactions between individuals and to analyze these interactions with techniques adapted from the field of social science in order to gain insight into the community and leadership structure of a "society"[1].

Figure 6 shows an example where we have tracked several subjects in the field of view of four PTZ cameras. Figure 7 shows an example where several individuals have been identified from zoomed-in PTZ cameras. Finally Figure 8 shows an example where an analysis of the interaction between individuals has revealed that there are two communities ("gangs") in this "society". An

[1]The leader in our context refers to the person that has most frequent contact with other group members. This definition has no implication of the person being the gang leader.

All Mock Prison Riot data was *enacted* by law enforcement and corrections practitioners.

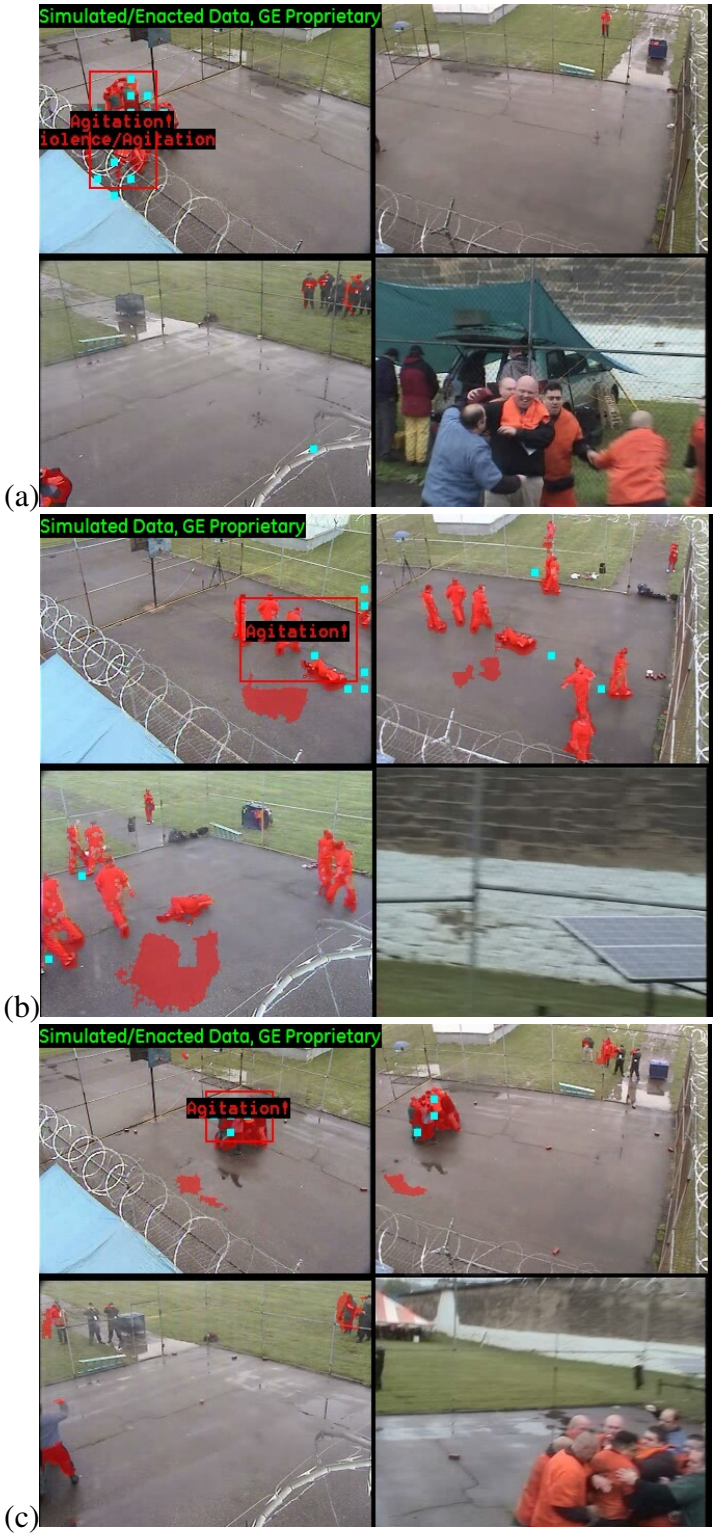


Figure 5: Aggression Detection. Three examples where the system detected a fight in progress. Every image contains red rectangles that outline the detected aggression region.

All Mock Prison Riot data was *enacted* by law enforcement and corrections practitioners.

interaction frequency also determined the most likely "leaders" of the communities. Figure 10 shows a larger social network estimated from a more busy scene shown in Figure 9.

Figure 6: Control of PTZ Cameras. Snapshots of the PTZ views, four views per snapshot in a row.

Details of the PTZ control and face capture system are presented in Section 6.1 and 6.2. The social network estimation and analysis is presented in Section 7.

In addition to the sophisticated framework for determining and analyzing social network structures, we also developed lower-level "gang" detection capabilities. This allows us to still leverage knowledge about the presence of multiple distinct groups in challenging environments such as prison yards, where the quality and duration of video observation might not be long enough to detect the fact that two or more distinct gangs are present in (for example) a recreational yard. Figure 11 shows an example from the Mock Prison Riot where the system detected the presence of two distinct groups shortly before a fight between two gangs broke out.

2.3 System Evaluation and Feedback

We performed a thorough performance evaluation based on the data collected at the 2009 Mock Prison Riot. The analysis indicates that our current system system has an about 70% chance of detecting the occurrence of disorderly or aggressive events in the observed prison environment and currently has a 20% chance of predicting the event *before* it occurs.

After every Mock Prison Riot data collection and testing session we solicited feedback from law enforcement officers with respect to the merit and performance of our proposed technology.

All Mock Prison Riot data was *enacted* by law enforcement and corrections practitioners.

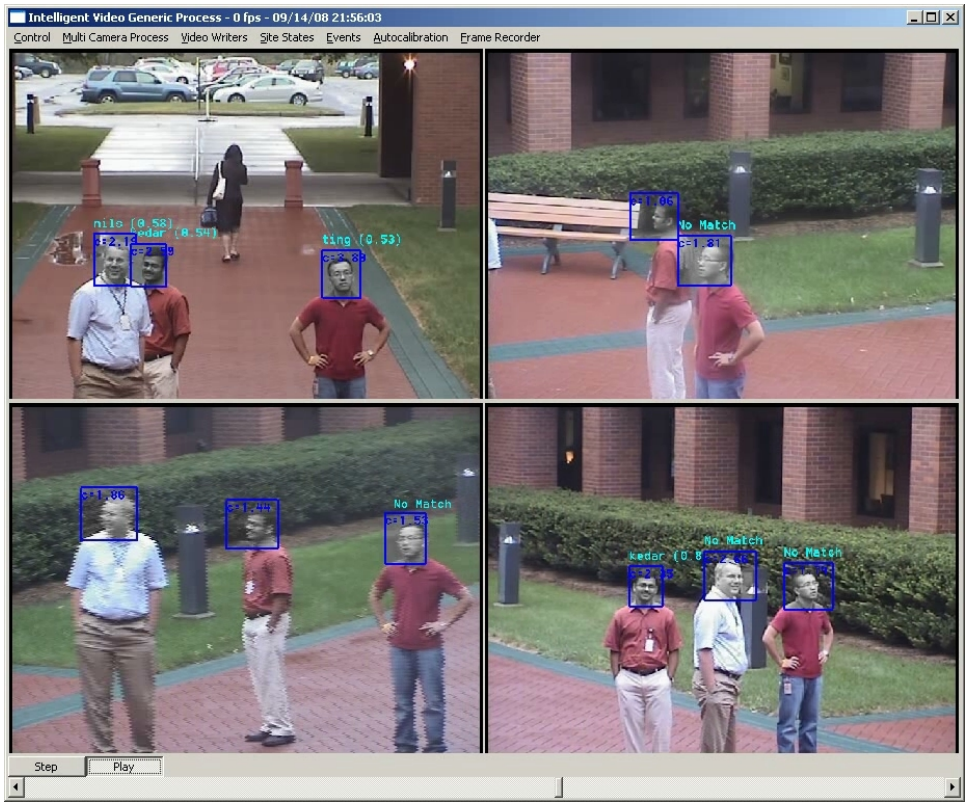


Figure 7: Face Detection and Face Recognition. Snapshot of the system performing face detection and face recognition. The blue rectangles denote locations where faces were detected in the PTZ views. Above the rectangles, names as estimated by the face recognition are shown. When "No Match" is shown, the face recognition could not determine a reliable identity estimate. When no name is shown, the face recognition system determined that no frontal face was present.

All Mock Prison Riot data was *enacted* by law enforcement and corrections practitioners.

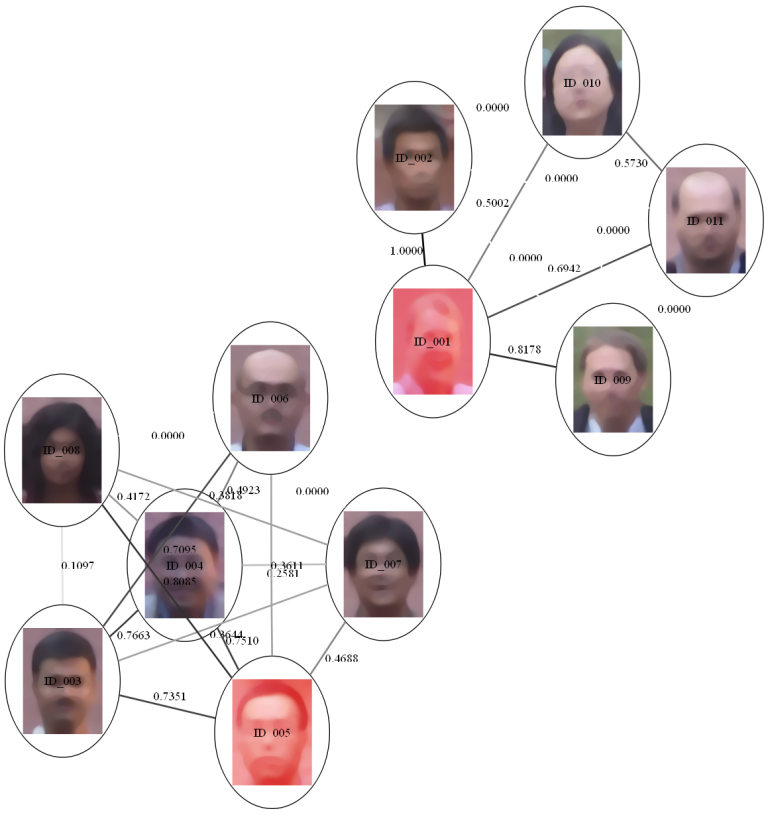


Figure 8: Leaders of Social Groups: In this experiment, there was a total of 11 individuals, divided into 2 groups. Members of each group appear in the scene at different times, during which the leader is always present. This generated, in a modularity sense, strong connections between the leaders of each group to its members. As a result, the system was able to identify the Eigen-leaders (Appendix E.2.3), which are the red nodes in this figure.

Figure 9: Large motion crowds: In this scene, there was a total of 23 individual subjects who were told to mingle in a 3-group configuration. The overlayed white wireframes with IDs show the tracking result from 4 fixed camera views.

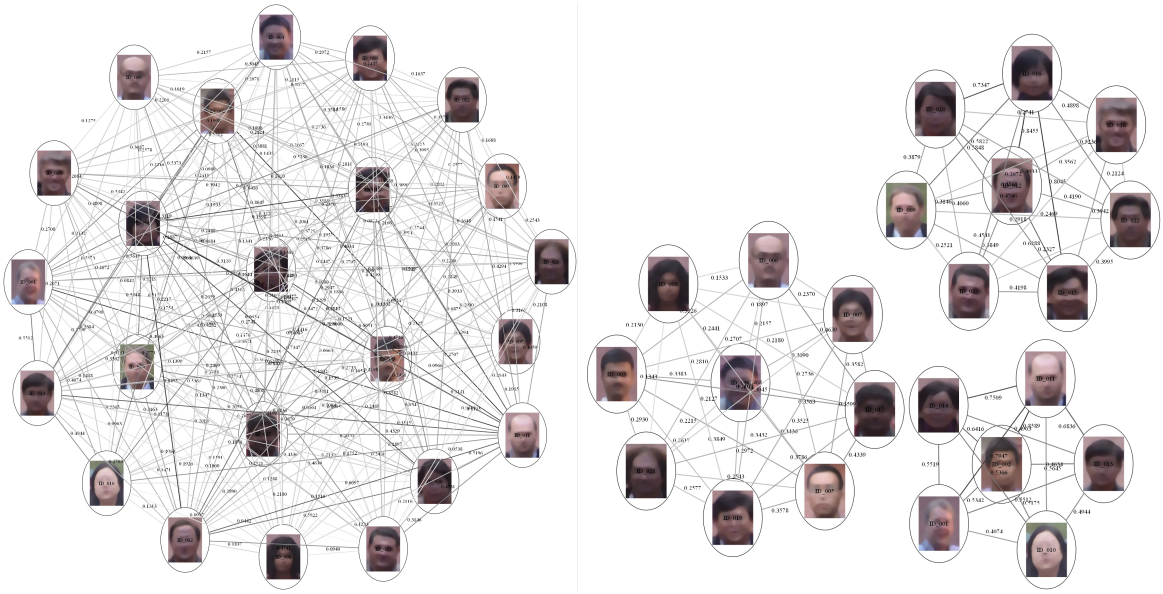


Figure 10: Discovering Social Groups: The left image shows the social network constructed from the sequence in Figure 40, which was divided into 3 groups correctly using the modularity-cut, shown on the right. Here, the links are shown with the weights between the individuals.

Figure 11: Distinct Group Detection. A detection algorithm has been developed that detects pairs of gangs that are stable and have certain spatial configurations with respect to one another. One group (red) is considered stable and loitering. It is considered to be distinct from the second group (green).

The general feedback was very positive and the correctional officers where enthusiastic about what the technology is able to accomplish in their operational environments.

Figure 12: Feedback. The GE Global Research team actively solicited comments and feedback after each scenario of the 2009 Mock Prison Riot. The response was overall very positive. Independent evaluation has also been conducted by the WVHTF in their "Mock Prison Riot Operational Assessment Report".

2.4 Law Enforcement Relevance and Impact

The following are a few examples of how the technology developed under this program can affect law enforcement operations and practice:

- The crowd parameter estimation (Section 5.3) has the ability to monitor public places and to alert authorities of large gatherings or when crowd density has reached a dangerous level.
- The motion pattern detection algorithms (Section 5.4) can automatically detect FOLLOWING and CHASING events, which can help to keep public parks safe and automatically detect suspicious events in correctional facilities.
- The aggression detection algorithms (Section 5.7 and 5.8) can automatically detect the occurrence of agitation and aggression. Aggression detection can be used to quickly dispatch officers to the scene of the event and also, if available, aim additional PTZ camera assets to obtain higher resolution footage of the event.
- The PTZ targeting system (Section 6.1) is a spin-off capability developed under this program. The algorithm was developed to enable the capture of faces for the social network

analysis, but is useful far beyond that particular application. Automatic PTZ targeting can be leveraged to capture faces (see Section 6.2) and high-resolution footage from a distance of subjects involved in any kind of suspicious activities without requiring operators to manually aim the camera. The developed system intrinsically strives for optimal capture strategies, which directly translates to potential savings in labor, time and equipment when compared to alternative ways to capture such imagery.

- Our research into detecting and recognizing faces from PTZ cameras has important applications for law enforcement as it describes methods for performing face recognition of uncooperative individuals from a distance.

- Our work is the first one to show that one can recover social network structure automatically from video. This can have an important impact on law enforcement. The social network analysis algorithm (Section 7) allows officers to automatically estimate social network structures and determine things such as group-size, membership and leadership structure automatically from video. The system has been shown to provide promising results and could be used in environments such as prison dining-halls, recreational yards and other environments, where law enforcement and corrections would like to automatically gather gang-intelligence and where such tasks nowadays have to be performed manually through careful observation.

2.5 Dissemination of Research Results

As part of this research program we have disseminated our work through the following papers:

[1] Nils Krahnstoever, Ting Yu, Ser-Nam Lim, Kedar Patwardhan, Peter Tu, "Collaborative Real-Time Control of Active Cameras in Large-Scale Surveillance Systems", Workshop on Multi-camera and Multi-modal Sensor Fusion Algorithms and Applications (M2SFA2), Marseille, France, October 18, 2008.

[2] Ting Yu, Ser-Nam Lim, Kedar Patwardhan, Nils Krahnstoever, "Monitoring, Recognizing and Discovering Social Networks", IEEE Conference on Computer Vision and Pattern Recogni-

tion, 2009.

[3] Nils Krahnstoever, Ting Yu, Ser-Nam Lim, Kedar Patwardhan, "Collaborative Real-Time Control of Active Cameras in Large-Scale Surveillance Systems", in "Multi-Camera Networks: Concepts and Applications", Editors: Hamid Aghajan and Andrea Cavallaro, Elsevier, 2009.

The following provisional patent was filed:

Nils Krahnstoever, Ting Yu, Ser-Nam Lim, Kedar Patwardhan, Peter Tu, "Collaborative real-time control of active cameras in large scale surveillance systems", Provisional Filing, US 61/091581.

2.6 Next Steps

In the follow up program "Advanced Behavior Recognition in Crowded Environments" (2009-SQ-B9-K013) we will focus on improving the behavior recognition of the system based on the capabilities and low-level motion pattern event detectors developed under this program. In particular we will focus on handling observation uncertainty better and to develop ways in which non-scientist operators can cooperate with the system to add new scenario and behavior recognition components to a system.

3 Introduction

In the rest of this document we will provide a comprehensive overview of our research program. We will first give an overview of the data sets that were collected and used as part of this study. In Chapter 5 we will describe our work on analyzing motion patterns of groups and crowds. Chapter 6 will describe our work on capturing and maintaining people's identity using face recognition in non-cooperative environments. This is a major technical component that enables the social network estimation work, which we will describe in Chapter 7. Chapter 8 will provide a detailed overview of our system testing effort at the 2009 Mock Prison Riot and our performance evaluation conducted on data collected at that event. We will conclude this report in Chapter 9. In the Appendices we provide detailed information about the list of papers published as a result of this research program, list all reviews and meetings as well as provide several tables listing details about the MPR data sets.

4 Data Sets and Data Collections

To evaluate the technology developed under this program we performed several data collections as well as used a number of existing and publicly available datasets. The first data collection was performed at the GE Global Research Center and featured group behaviors and motion patterns enacted by GRC scientists. Later during the program, additional data collection and testing was performed at the 2009 Mock Prison Riot (MPR) in Moundsville, WV.

4.1 GE Global Research Collection

Figure 13 shows images from a data collection at the GRC testbed where behaviors such as group movements, group chasing, following, group formation and dispersion, fallen individual have been enacted. The data was collected from four synchronized video cameras.

All Mock Prison Riot data was *enacted* by law enforcement and corrections practitioners.

Figure 13: Data Collection of Movement Patterns. Data collection at GRC testbed. All scenes are enacted by GE Global Research Employees.

All Mock Prison Riot data was *enacted* by law enforcement and corrections practitioners.

4.2 Airport and "Behave" Data

In addition to data collected at GE Global Research, we utilized publicly available and existing datasets for this study. Figure 14 shows an example of a fight between a group of three people in a sequence from the BEHAVE dataset. Figure 15 shows a sequence from an airport terminal exit.

Figure 14: "Behave" Dataset. Example frames from a publicly available dataset we will utilize for this study. From http://groups.inf.ed.ac.uk/vision/behavedata/interactions/index.html

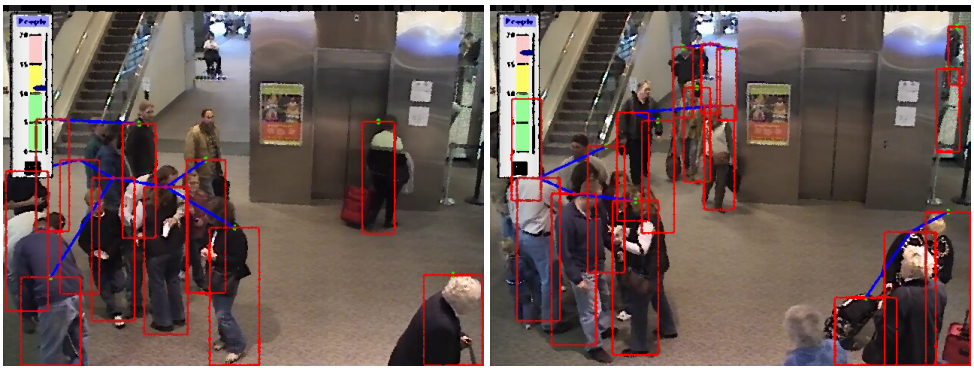


Figure 15: "Airport" Dataset. Existing dataset from an airport departure lounge that allows the study of crowd level processing, especially crowd size estimation and crowd formation.

4.3 Mock Prison Riot Data

Based on the recommendation from NIJ we utilized the Mock Prison Riot (MPR) event, sponsored by NIJ and organized by the West Virginia High Technology Consortium Foundation (WVHTCF) to perform an extensive data collection and system evaluation. The MPR data collection was performed toward the end of the program. The collection was very successful and it is planned to utilize this venue again on future NIJ programs.

All Mock Prison Riot data was *enacted* by law enforcement and corrections practitioners.

The MPR is held annually on the grounds of the decommissioned West Virginia Penitentiary in Moundsville. The goal of the Mock Prison Riot is to enable law enforcement and corrections (LEC) officers to perform tactical training exercises and provide exposure to new technologies. During the MPR, LEC officers traditionally practice how to handle various out-of-order events in prisons, such as riots, fights, and hostage situations. During these exercises, law enforcement and corrections officers enact the activities of prisoners. The MPR event is an ideal venue for having officers enact realistic prisoner behaviors for testing and evaluation purposes.

4.3.1 Venue

The MPR takes place at the decommissioned West Virginia Penitentiary in Moundsville. The penitentiary grounds consist of several large cell blocks attached to several outdoor recreational yards. Figure 16 shows an aerial view of the prison and two of the main yards where many of the MPR scenarios are carried out.

(a)

(b)

(c)

Figure 16: West Virginia Penitentiary Yards. (a) aerial view of the prison grounds, (b) recreational yard, and (c) large outdoor yard.

All Mock Prison Riot data was *enacted* by law enforcement and corrections practitioners.

It was envisioned to perform data collection and system testing at multiple locations throughout the venue but during the event it became apparent that the complexity of the system setup (to be described below) in addition to the weather conditions (at times heavy rain) made moving the system infeasible and the exercise yard (bottom left in Figure 16) was chosen as the one central venue for testing and data collection.

4.3.2 Installation

Data collection was performed with a total of four synchronized video cameras:

- two high resolution GE Legend PTZ cameras mounted on a mobile tower (tower provided by WVHTCF)
- one high resolution GE Legend PTZ mounted on a large broadcast quality tripod
- one Sony EVI-D70 PTZ camera mounted on the fence inside the recreational yard

The equipment for processing and recording the video data was stored in a mini-van. The equipment was accessed from a table outside of the van, which had to be covered by a tarp due to heavy rainfall during the first two days. This setup proved to be non-optimal both due to the rain and also due to bright sunshine, which made it difficult to see information on the computer screens. At future collections it would be desirable to place the recording equipment, keyboard and screens inside of a mobile trailer or similar enclosed shelter, which would also make it easier to demonstrate the technology to law enforcement practitioners.

4.3.3 Camera Views

From the placed cameras we were able to capture activities from both inside the recreational yard as well as from the large lawn area north of the recreational yard (see Figure 19).

(a) 

(b) 

(c) 

(d) 

Figure 17: Cameras. (a) mobile tower with mounted cameras, (b) the left-most and right-most cameras belonged to the GE Global Research system, (c) broadcast tripod with mounted camera, and (d) computer controlled PTZ camera, mounted on fence.

Figure 18: Equipment. The processing and recording equipment was housed in a van, with screens and keyboard placed on an outside table.

4.3.4 Calibration

The GE Global Research system requires accurate calibration of all cameras with respect to a common world coordinate reference frame. We utilized a combination of natural landmarks such as fence posts as well as manually supplied ground plane reference points to calibrate the cameras. It is common for this process to take some time but bright sunlight made it difficult to see the content of the computer screens that where designed for indoor operation. This slowed down the calibration process. It took about three hours to calibrate the recreational yard and one hour to calibrate the lawn area north of the recreational yard. On separate research efforts, GE Global Research is continuously improving the calibration procedures and the time for setup and calibration will be drastically reduced in the future.

Further data about the Mock Prison Riot data collection will be provided in Section 8 of this report.

(a)

(b)

Figure 19: Camera Views. (a) four camera views of the recreational yard and (b) four camera views of the north yard.

5 Motion and Crowd Pattern Analysis

In this section we will describe our approach to analyzing motion patterns in crowded environments. The capabilities where first developed based on data collected at the GE Global Research Center and later refined based on the data and experiences gained at the Mock Prison Riot 2009.

5.1 Multi-camera Multi-target Tracking

The analysis of crowd motion patterns requires the reliable detection and tracking of individuals in the view of one or more surveillance cameras. This is enabled by the existing GE Global Research

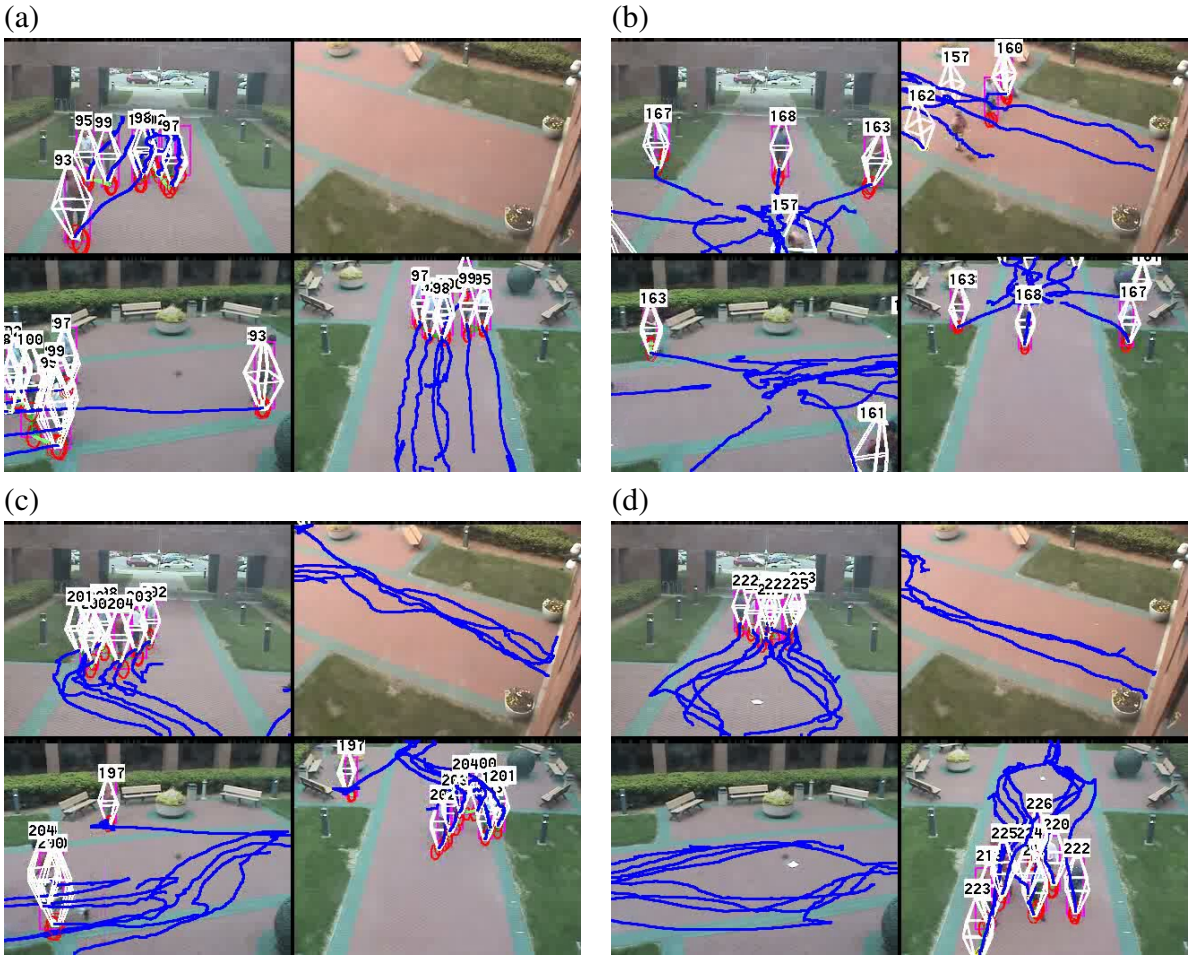


Figure 20: Tracking Results. Multi-camera multi-target tracking system is applied to collected test data. Shown are scenes depicting (a) chasing, (b) group dispersion, (c) group avoiding subject, and (d) group avoiding object.

All Mock Prison Riot data was *enacted* by law enforcement and corrections practitioners.

intelligent video technology that encompasses comprehensive capabilities in the areas of camera calibration, crowd segmentation, detection, and tracking. The employed tracker performs multi-camera multi-target tracking in a centralized, calibrated environment, where the estimated state of each person, i.e., the location, size and velocity of each individual, is defined in a metric coordinate system. Several sample images showing some representative tracking results obtained by the system are illustrated in Figure 20. The system performs centralized tracking, i.e., the identities of targets are maintained across camera views. Provided that each individual has been tracked on the ground plane by the tracking algorithm, the system obtains trajectories for all individual targets.

5.2 Detection and Tracking of Motion Groups

The first step towards the crowd motion pattern analysis is to detect and track the motion groups over time. We define a motion group as a set of targets moving together in a spatially and temporally consistent way, where consistency is measured by a pre-defined distance metric (see below). Such a distance metric could be defined between individual targets based on the target state trajectory data as well as between individual targets and hypothesized motion groups. These two methods of computing the distance metric essentially lead to two different ways of dealing with this detection problem of motion groups. The first one, i.e. defining the distance metric between individual targets, essentially resorts to a data-driven approach, where the data here refers to the state trajectory of individual targets; while the second one, i.e. computing the distance metric between targets and hypothesized motion groups, is intrinsically a model-based method, where the model naturally refers to the hypothesized motion groups.

In this work, we chose the data-driven approach to solve the detection and tracking problems of motion groups. The procedure of applying a data-driven method can be decomposed into the following steps: data clustering for motion group detection, and temporal association for motion group tracking.

Given a surveillance site, we assume that our multi-camera multi-target tracking system observes M targets at time t. Each target i, where i represents the target identifier and $i \in \{1, \ldots, M\}$,

has a motion state vector $x_{i,t}$ describing the ground plane location $x^l_{i,t}$ and velocity $x^v_{i,t}$ of the target i at time t. The complete state history up to time t, i.e., target trajectory, is denoted as $\mathbf{x}_{i,t} = \{x_{i,1}, \ldots, x_{i,t}\}$.

Given a pair of targets, we define a pairwise distance function $\mathbf{D}(i,j)$ based on their trajectories $(\mathbf{x}_{i,t}, \mathbf{x}_{j,t})$

$$\mathbf{D}(i,j) = f(\mathbf{x}_{i,t}, \mathbf{x}_{j,t}) \tag{1}$$

Such a distance function measures the spatial-temporal consistency of two involved targets and estimates how likely they move along with each other, i.e., the possibility of two targets moving in the same motion group. Both the current relative locations of two targets and their direction of travels should be considered for the similarity. If two targets from two different groups happen to come across each other at the current time t, though their relative locations are close, they should not be treated as a similar pair since historically they are from two different directions. On the contrary, even if two targets may share similar direction of travels during the past several seconds, but if their relative locations are far from each other at the current frame, they also should not be considered as a good pair. Hence, our distance function $f(\mathbf{x}_{i,t}, \mathbf{x}_{j,t})$ is defined by combining these two similarity quantities as follows

$$\mathbf{D}(i,j) = \exp^{-\frac{\|x^l_{i,t} - x^l_{j,t}\|^2}{2\sigma_l^2}} \exp^{-\frac{\sum_{\tau=t-\Delta t}^{t} \|x^v_{i,\tau} - x^v_{j,\tau}\|^2 / \Delta t}{2\sigma_v^2}} \tag{2}$$

where the first term measures the relative distance of two targets at the current time t, and the second term measures the direction of travel consistency in the past Δt seconds. Evaluating this quantity for all pairs of individuals in the site defines a weighted graph $G_t = (V_t, E_t)$, where the vertex set V_t contains all individual targets $\{\mathbf{x}_{1,t}, \ldots, \mathbf{x}_{M,t}\}$, and the edge $E^{i,j}_t$ in the weighted edge set E_t is assigned the value $\mathbf{D}(i,j)$. Given such a graph, different graph partitioning methods could be applied to identify the current groupings of individuals, such as Normalized Graph Cut [2] or Minimal Spanning Tree [3]. For simplicity, at this stage, a Minimal Spanning Tree approach is adopted to find out all sub-graphs that can be clustered and deemed as independent motion groups.

A Minimal Spanning Tree is a sub graph without cycles that connects all the vertices together and that has minimal total edge weight. A threshold can then be set on the edge weights of this Minimum Spanning Tree to prune away the edges whose weights, measuring the spatial-temporal motion consistency of two incident target nodes, are below the threshold. The isolated sub-trees after pruning form the groupings of individuals, i.e., the detected motion groups.

The chosen graph partitioning algorithm, specifically the Minimal Spanning Tree algorithm currently used in our system, clusters the individual targets into different motion groups in a frame-by-frame basis. However, it has no notion about the correspondences between the motion groups obtained from the last time instant and the ones clustered at the current time instant, i.e., the identifiers (ID) of the motion groups are not carried over time. To bridge this gap, a temporal data association at the group level is applied. The principle to assign each motion group in the current time instant to their previous counterparts is based on the majority-voting criterion. Since each individual target has its own unique target ID, when these targets propagate over time and become grouped, we can count the number of overlapping individuals that two motion groups from two successive times have shared. Thus, we assign each of the motion groups in the current time with the ID of a previous motion group that the current one has the maximal overlaps (majority voting). Simultaneously, new groups that have not been assigned to any previous groups will receive a new unique group ID, and old groups that do not correspond to any newly clustered groups will be terminated.

Once these basic capabilities of detecting and tracking crowded motion groups are achieved, the questions become how the system can leverage them to immediately achieve some group-level semantic understanding of behaviors In the following, the crowd parameter estimation results, and simple event detections obtained so far are presented.

5.3 Counting and Crowd Detection

The output of the tracking system directly yields an estimate for the number of people visible in a scene, which can in turn be used for the detection of crowded and congested conditions. The

detection and tracking of motion groups yields a decomposition of a crowded scene into groups, where group size and group density can furthermore be used for detecting a range of crowd level events. Figure 21 shows the results of our system applied to an airport scenario and a graphs that shows the estimated number of people over time.

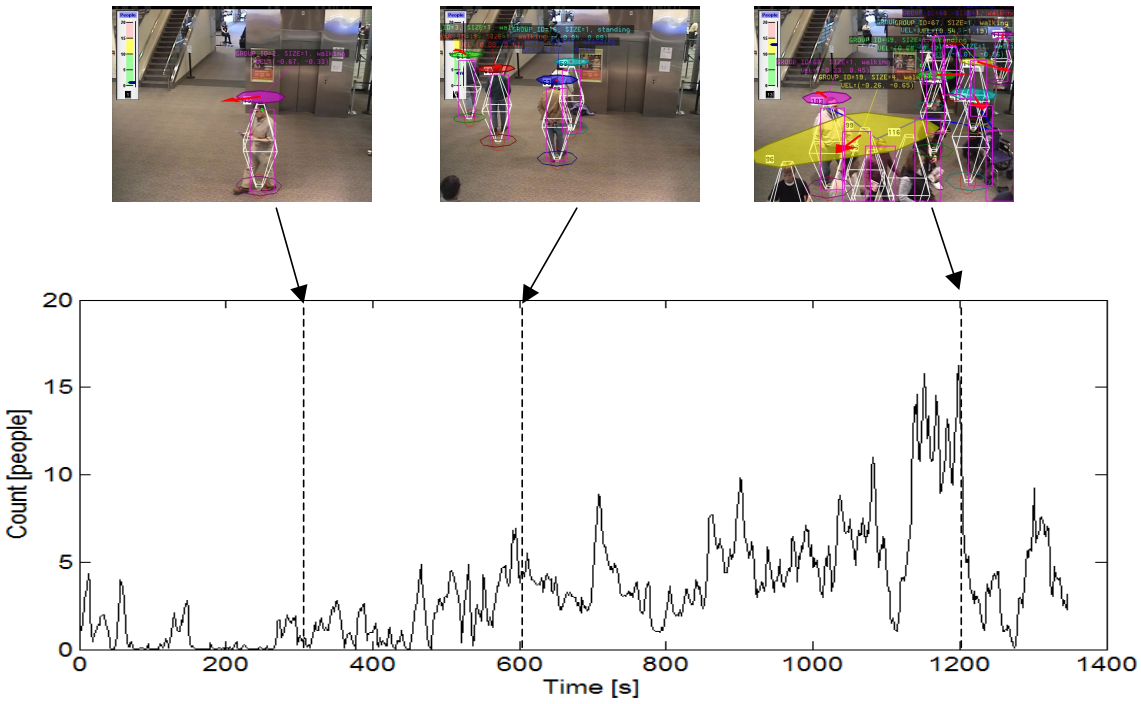


Figure 21: People Counting in Crowded Conditions. Crowd detection and counting applied to an Airport Dataset.

5.4 Simple Group-Level Events

Once the motion groups are detected and tracked, several basic group-level attributes can be computed, such as group location and velocity at every time instant. Group status, including whether the group is in a standing, walking or running mode, can be defined based on these attribute quantities. Simple group events, such as group following and chasing, can be detected based on the status of individual group and their interactions. Here, we define the event FOLLOWING as the case that one group is following the other group by having similar directions of travel, and both of them are considered to be WALKING. Similarly, we define a CHASING event as two groups following each

other, while RUNNING. Figure 22 shows one example of a CHASING event detected by the system. Additional group cues, which also apply to individuals, are FAST MOVER, which indicates a fast individual or group.

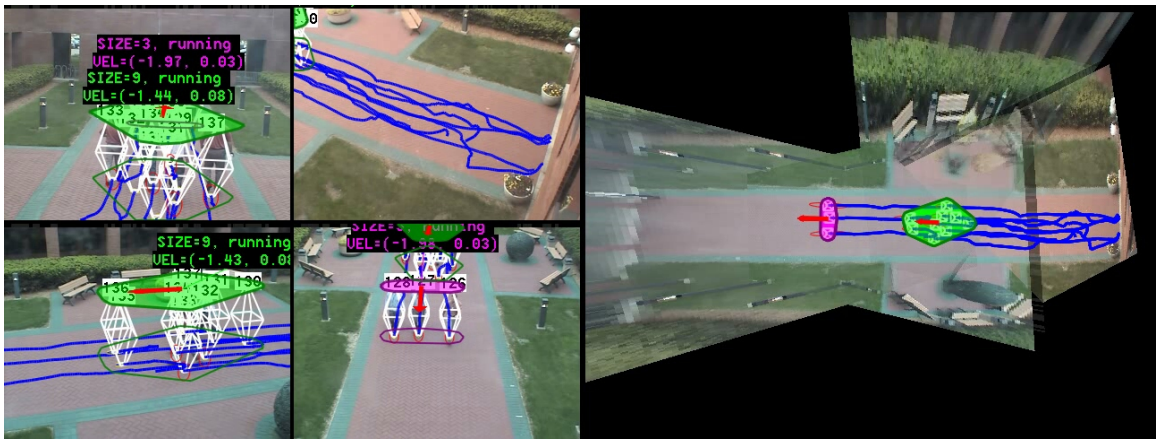


Figure 22: Crowd Motion Pattern - CHASING. Visualization of the grouping and crowd analysis stage. A larger group chases a small group. Both are running. The colored regions visualize the clustering result. The right side of the display shows a virtual top-down visualization of the surveillance site. Red arrows indicate the speed and direction of travel.

5.5 Group Interaction Model

In typical crowd scenes, people move and interact with others very frequently, which leads to the tracked motion groups consistently changing their memberships. For example, it is very common to observe that an individual target, which may previously belong to one motion group, later departs from this group and joins the other motion group. Such a transition caused by individual targets induces a temporal interaction between the involved motion groups. Our system captures these frequently seen interactions by building and continuously expanding a group interaction model over time.

Considering an illustrative example in Figure 23 (left), there are 6 individual targets $\{\mathbf{x}_1, \ldots, \mathbf{x}_6\}$ that move in the scene across three time instants, t_1, t_2, t_3. At t_1, the system detects three motion groups, group $\{A_{t_1}, B_{t_1}, C_{t_1}\}$, where $A_{t_1} = \{\mathbf{x}_{1,t_1}, \mathbf{x}_{2,t_1}, \mathbf{x}_{3,t_1}\}$, $B_{t_1} = \{\mathbf{x}_{4,t_1}, \mathbf{x}_{5,t_1}\}$, and $C_{t_1} = \{\mathbf{x}_{6,t_1}\}$. When these motion groups evolve to time t_2, group A is split into two sub-groups,

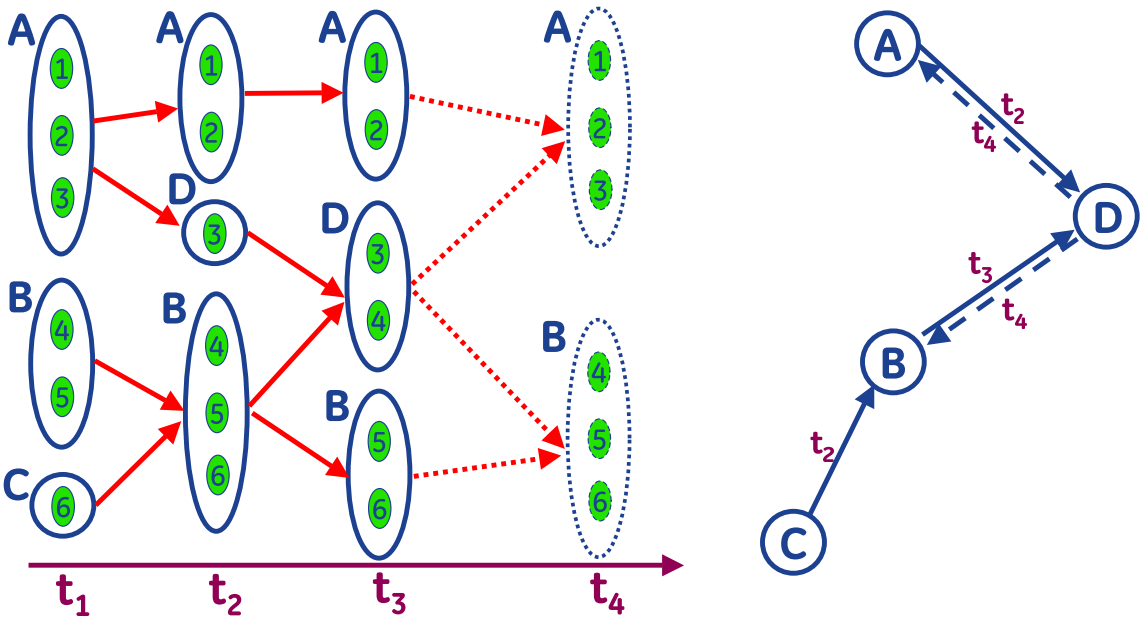


Figure 23: Group Interaction Graph. Temporally evolving interaction graph (left) and interaction model (right).

one of them inherits the same group ID and becomes group A_{t_2}, while the other one receives a new group ID D_{t_2}. At the same time, group B and C merge into a single group, which retains the ID of group B due to majority voting criterion and becomes group B_{t_2}. In analogy, from time t_2 to t_3, group B splits into two parts, and one part merges with group D and become group D_{t_3}, and the other part remains as group B_{t_3}. Such temporal interactions among motion groups could be characterized by a graph-based interaction model as shown in Figure 23 (right), where each node represents one group, and each edge denotes a temporally causal relation. For example, group C merges into group B at time t_2, thus there is an edge from C to B, denoting C is a parent of B and B is a child of C. In our case, each edge not only describes the relation between two involved groups, but also records the time when this interaction takes place. In the above example, the edge from group C to group B will have a time tag t_2, denoted as $e^{t_2}_{C,B}$. Similarly, group A and group B both split, and parts of them become or merge into group D, so in the generated graph interaction model we have two edges, one from group A to group D, and the other from group B to group D. But they will have two different time tags, i.e., $e^{t_2}_{A,D}, e^{t_3}_{B,D}$, as illustrated in Figure 23 (right).

With the time evolvement, such a generated graph will become very complex when more groups start to interact or new groups emerge. What makes the problem more complicated is that two groups could become mutual parents. As shown in the dashed parts of Figure 23 (left), at time t_4, individuals in group D are split into two parts, and join group A and B separately, making group D also become the parents of A and B, shown as the dashed edges in Figure 23 (right). Fortunately, the time tags on the edges can be used to differentiate these different cases. In fact, in this proposed graph interaction model, it is possible that there exist multiple directed edges between two group nodes. It is the introduction of time tag along each edge to make such a graph construction possible.

5.6 Group Formation and Dispersion

As such an interaction graph is continuously expanded over time, interesting and complex group-level events can be detected. We have implemented two events based on this generated graph – GROUP FORMATION and GROUP DISPERSION. A GROUP FORMATION event can be vaguely defined as several individuals coming together to meet each other from different places and forming a large motion group. Vice versa, a GROUP DISPERSION event, can be defined as a large motion group gradually splitting into several small groups over time.

Consider the example in Figure 24 (left), which shows a typical group dispersion event. Six people in group A gradually split into individuals, generating several small motion groups. The corresponding interaction graph is shown in Figure 24 (right). It is naturally seen that if the system starts with node A and conducts a breadth-first-search (BFS) to count how many descendant nodes group A has within a temporal window (which is achievable due to the time tag along each edge), in this way we can decide if a group dispersion has happened by comparing the number of descendants with a pre-defined threshold. Sample frames showing the detection of this group dispersion event are illustrated in Figure 25. In analogy, a group formation event can be detected in a similar way, but counting how many ancestors one node has within a temporal window.

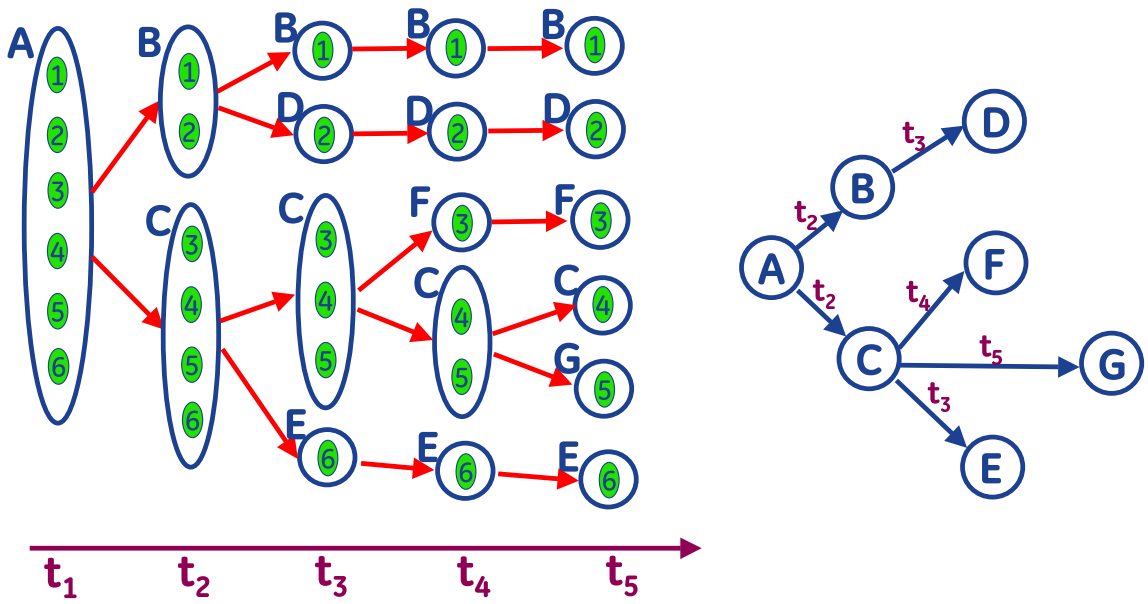


Figure 24: Group Interaction Graph. A graph resulting from a dispersion event.

5.7 Agitation and Fighting

One important behavior cue is *agitation level*, which is a predictor and indicator for conflicts, violence or fights. We define agitation to be unusual motion by a person or group in one place in space. In other words, an agitated person or group is one that remains at the same location, but still exhibits motion in the image. We measure agitation via the average velocity magnitude of foreground interest points at the places in the image where people are located. In the first implementation we used the Lukas-Kanade-Tomasi Feature Tracker [4, 5], constrained to only track foreground interest points, to estimate sparse optical flow. After grouping, feature tracks within all regions of interest that are induced by the individuals that constitute the group are used measure the average absolute velocity magnitude. If the velocity magnitude exceeds a pre-determined threshold *and* the group is considered to be stationary, we view the groups as agitated. Figure 26 shows a group of people involved in a fight, with the system having correctly identified this event.

Figure 25: Crowd Motion Pattern - Dispersion. Members in a group (top) are fleeing from each other (bottom), possibly triggered by an unusual event.

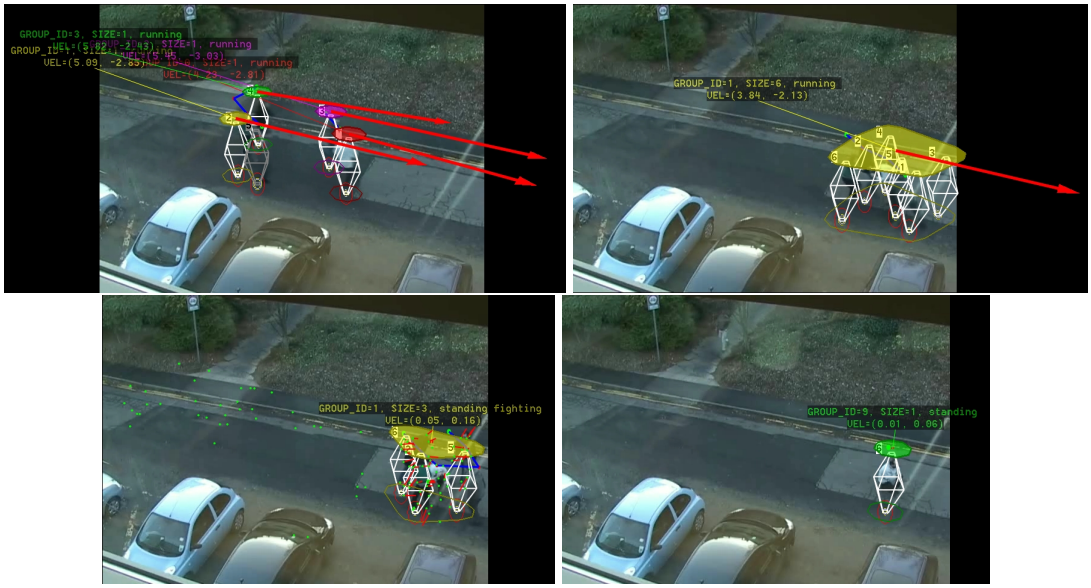


Figure 26: Crowd Motion Pattern - Running and Fighting. Four people running and subsequently fighting. Note the 'fighting' attribute associated with the group in the bottom left image.

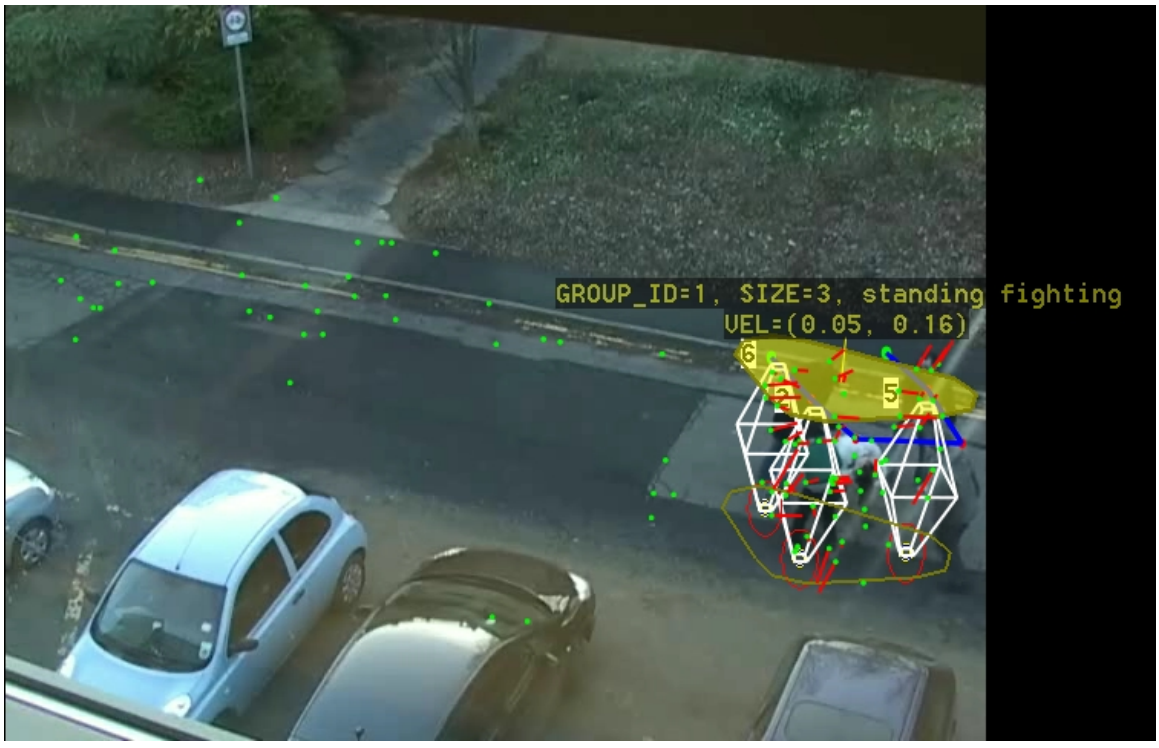


Figure 27: Detection of Fighting. The event 'fighting' detected in the BEHAVE dataset.

5.8 Advanced Aggression Detection

In response to the experience gained from the Mock Prison Data, we added and improved several behavior recognition components in our system. In particular the aggression detection components where improved considerably which will be described in the following section:

The previously developed aggression detection was found to consume too many computational resources and to provide sub-optimal detection results. We hence took a new approach to aggression detection, which improved both computational requirements as well as provide better detection performance. Unlike the previous approach, the current method does not attempt to detect aggression on a person by person basis, but rather attempts to decide on a region-by-region basis whether or not a degree of aggression (or rather agitation) is observed in the image. To that end, the video frame is divided into small "decision blocks" of size 16×16 and for every block a decision is made whether or not agitation is observed inside of this block.

5.8.1 Feature Tracking

Feature tracking is the process of detecting and tracking points in the image that have a high texture or "corneredness". Feature point tracking is an important pre-cursor to obtaining information about the movements of targets. The previous aggression detection approach was based on analyzing sparse optical flow using the Kanade Lucas Tomasi (KLT) feature tracker [6, 5]. This algorithm, although quite reliable, is very slow and was hence replaced with a new motion tracking approach. We now utilize the FAST feature detector developed by Tom Drummond [1]. The approach uses a heuristic to determine high-textured points but is extremely fast compared to other approaches. To obtain trajectories of detected points, we developed a data association point feature tracker. The feature tracker provides the detections $\mathbf{o}_i = (x_i, y_i)$ in image coordinates over time from which trajectories of feature targets are formed. The state of every feature target is represented by a four dimensional state space vector $\mathbf{x}_j = (x_j, y_j, vx_j, vy_j)$ in image coordinates. At every time step targets $\mathbf{x}_j$ are associated with detections $\mathbf{o}_i$, i.e., it is decided based on appropriate distance measures, if $a_{ij} = 1$ for a target i and observation j. After data association, the state of all

targets is updated according to the newly assigned observation, using a linear Kalman filter. For computational reasons the traditional association step based on the Hungarian algorithm was replaced by a greedy search strategy that in addition uses geometric binning to speed up the distance computation and nearest-neighbor search. Figure 28 shows the feature tracking sub-component in action.

Figure 28: Fast Feature Tracking. Example screen shots that show the newly developed feature tracking. The system tracks the motion of interest points and visualizes them as blue trajectories. For computational reasons only foreground regions (in red) are considered for tracking.

5.8.2 Motion Analysis

After the tracking step, every trajectory is analyzed with regards to a range of motion attributes and the attributes are carried over into the "decision blocks" mentioned above. The maintained attributes include:

- **Discontinued Tracks**: an agitated person or group is likely to rotate and change position rapidly in the image. Hence features will rapidly appear and disappear in the image and the associated tracks will discontinue. We measure the number of discontinued tracks per block as a measure of agitation.
- **New Tracks**: along the same argument as above, new tracks are an indication of agitated motion.
- **Erratic Motion**: agitated motion is viewed to be irregular and hence difficult to predict. We directly measure this aspect by accumulating the prediction errors over time.
- **Non-Stationarity**: we require that agitated motion does not remain stationary around a fixed location. This removes motion responses caused from e.g. trees blowing vigorously in the wind.

If multiple trajectories fall into a decision block, the data from multiple trajectories is combined probabilistically. For example, a single trajectory, say o_k, provides us with evidence that a certain block contains *erratic motion*. We denote this event with $\mathrm{E}_i = \{\text{Block } i \text{ contains erratic motion.}\}$ and the probability of erratic motion due to the observed trajectory as $p(\mathrm{E}_i|o_k)$. The probability of the block *not* containing erratic motion is then

$$p(\neg \mathrm{E}_i|o_k) = 1 - p(\mathrm{E}_i|o_k) \tag{3}$$

If we observe multiple trajectories $\mathbf{O}_i = \{o_k | o_k \text{ in Block } i\}$, we have that

$$p(\neg \mathrm{E}_i|\mathbf{O}_i) = \prod_{o \text{ in } \mathbf{O}_i} (1 - p(\mathrm{E}_i|o)) \tag{4}$$

and hence

$$p(\mathrm{E}_i|\mathbf{O}_i) = 1 - \prod_{o \text{ in } \mathbf{O}_i} (1 - p(\mathrm{E}_i|o)) \,. \tag{5}$$

The estimate $f_{\text{erratic}} = p(\mathrm{E}_i|\mathbf{O}_i)$ is part of a multi-dimensional feature vector that is maintained

on a per decision-block basis. Other features (non-stationarity etc.) are computed similar to the approach outlined here.

5.8.3 Motion Classification and Clustering

The per-block features are then classified according to a learned agitation model. We utilize a Support Vector Machine trained on a small number of example sequences to obtain an optimal classification. Training data was obtained by manually marking regions in some training sequences where agitation/fight occurred and collecting all feature vectors from these training regions as positive samples. Vectors outside of the marked regions where used as negative samples. Following classification, all blocks that where classified as agitated are clustered using an agglomerative clustering algorithm into larger agitation regions. Agglomerative clustering works by iteratively merging blocks together if the distance between regions falls below a certain threshold. We use Euclidean distance as the distance measure and use an element-to-element distance of $d_{\max} = \sqrt{6}$ block widths.

5.8.4 Results

The new agitation detection process has a better run-time performance and detection rate than the previous method. First quantitative experiments have shown that the detection rate is around $DR = 0.85$ and the false alarm rate $F = 0.50$ (observed false alarms per true detection). Missed detections occur in situations where the aggression is a quick stabbing and not much struggle/fighting occurs before and after the event. False alarms are caused by issues such as basketball play (which in some sense is not incorrect), actors hitting the fence, which causes the entire camera equipment to shake vigorously and activities that give the appearance of agitation. Figure 29 show several example detections from data captured at the Mock Prison Riot. The system is reliably picking out aggression detection events. Figure 30 shows a false alarm caused by the prisoners leaving the recreational yard through a tight entrance door.

Figure 29: Aggression Detection. Nine example instances where the system detected a fight in progress. Every image contains red rectangles that outline the detected aggression region.

Figure 30: Aggression False Alarm. False alarm caused by actors leaving the recreational yard.

6 Identity Management

Crowd pattern analysis involves the study of individuals and groups and behaviors that can be gleaned from their motion pattern. As a next step, we extended this analysis to a semantically higher level by maintaining a long-term record of people's *identity* in addition to their motion data. This is in one sense a classical biometrics challenge. The following issues had to be tackled: (i) We had to capture and maintain biometric information of "non-cooperative" individuals, i.e., individuals that are not aware of participating in (or posing for) a biometric enrollment or verification process; (ii) we have to capture information visually from a distance; (iii) individuals can be moving and move in groups and crowds.

6.1 PTZ Camera Control

As a first step, we devised a strategy to capture high-resolution facial shots of individuals that are in the capture zone of multiple PTZ cameras.

This section has been adapted from a self-contained technical paper written for this program. The original paper was published at the Workshop on "Multi-camera and Multi-modal Sensor Fusion Algorithms and Applications" and was entitled "Collaborative Real-Time Control of Active Cameras in Large Scale Surveillance Systems" by Nils Krahnstoever, Ting Yu, Sernam Lim, and Kedar Patwardhan.

The goal of PTZ camera control is to acquire close-up imagery of people in a surveillance site. The control algorithm is driven by the output of a multi-camera, multi-target tracking system that performs person detection and tracking from a set of fixed surveillance cameras. This section addresses the problem of optimally scheduling the PTZ cameras in real-time under a variety of performance objectives. The scheduler attempts to choose an optimal plan that maximizes the probability of successfully completing a biometrics task. We present a novel objective function that balances the number of captures per target and the quality of captures. We will also show the results of our system operating in real-time under real-world conditions on four PTZ and four fixed

CCTV cameras, all controlled via a single workstation.

6.1.1 Introduction

Automatic surveillance systems usually operate with fixed CCTV cameras as this allows the use of efficient detection and tracking algorithms. Unfortunately the resolution of even high-quality CCTV cameras is at most D1 (720x480), but often much lower. This makes subsequent tasks that are dependent on the captures, such as face detection, face recognition, and forensic examinations, difficult, especially if subjects were imaged from a significant distance. Mega-pixel sensors have the potential of overcoming such a resolution problem, but their real-world deployment is still very limited, mostly due to the challenges associated with managing, processing and recording the non-standard imagery from such sensors. On the other hand, PTZ cameras are used extensively for security applications. PTZ cameras can be used to obtain close-up views of subjects and their activities, and it is natural to combine the autonomy of fixed camera systems with the versatility of PTZ cameras, where detection and tracking performed by fixed cameras can be used to control PTZ cameras. For example, in [7], a single PTZ camera is "slaved" by a fixed camera for face detection. With large-scale multi-camera surveillance networks becoming more prevalent, the problem of scheduling and controlling multiple PTZ cameras to monitor activities has to be addressed.

A PTZ camera scheduling strategy has to choose a control plan that meets several intuitive requirements: (i) no individual target should be neglected, (ii) no individual should receive excessive preference, (iii) the quality of captures should be optimized, (iv) the assignment of PTZ cameras to targets should be dependent on their capacities to meet the quality requirement of captures. We have developed a formalized method that attempts to meet the requirements. More specifically, we consider high-quality close-up captures of individuals for biometrics tasks. Towards this end, a novel objective function that characterizes the overall probability of successfully completing these biometrics tasks is defined. We introduce a set of capture quality measures to quantify the aforementioned success probability. The proposed objective function considers the assignments of all PTZ cameras to targets jointly, and the optimization of this objective results in an interesting

scheduling strategy that controls a set of PTZ cameras to work collaboratively towards obtaining high-quality imageries of all site-wide activities.

In the following, we firstly discuss the related work in Section 6.1.2. We report the experimental results in Section 6.1.3. Some discussions are made in Section 6.1.4. The technical expositions and further details are in the Appendices D.1, D.2, D.3.

6.1.2 Related Work

The primary literatures on multi-camera automatic surveillance systems address the issues related to low and middle-level vision processing, such as person detection and tracking [8, 9, 10, 11], multi-target and multi-camera tracking in a site-wide manner [12, 13] to mention a few. The extracted tracking data can be utilized to drive a set of PTZ cameras to acquire high-quality imageries and to, for example, build biometrics signatures of the tracked targets autonomously. The capturing task requires not only the (auto)calibration between the fixed and PTZ cameras to establish the correspondence of coordinate systems [14], but also the planning of camera scheduling strategies to assign the PTZ cameras to interesting targets and guide their pan-tilt-zoom movements [15].

The simplest setting of autonomous target capturing is to use a master/slave camera system configuration, where the applied strategy of controlling the slave PTZ camera is usually based on some easily predefined heuristics [16, 7]. The camera scheduling becomes non-trivial, when the system attempts to simultaneously capturing multiple moving targets with fewer PTZ cameras. A number of different camera scheduling policies were designed in [17] driven by the application goals. For example, a round-robin scheduling is developed to periodically assign cameras to different targets which achieves a uniform capture of all targets with close-up views. Rather than giving the equal importance to each target, [18] proposed to order the moving targets in a priority queue based on their arrival time and number of times with that the target has been captured. The set of currently available PTZ cameras are also ranked based on easiness of adjusting their PTZ states to capture the target with the top priority, and the camera with the top rank is chosen to finish the capture task. [19] ranked the targets by the deadlines when they will leave the surveillance area. An

optimal subset of the targets, which satisfies the deadline constraints, is found through an exhaustive evaluation. However, they schedule the capture task for each PTZ camera independently. In crowded surveillance scenes, moving targets may show significant interactions, and for some moment one target may be occluded by others from the camera view. [20] proposed to estimate these occlusion moments for each target based on the camera geometry and target motion prediction, and constructed the visibility intervals for each capture task, which is defined as the complement of occlusion moments. Based on these visibility intervals, they solve the task scheduling problem using a greedy graph search method [21].

Though bearing some similarity to our proposed approach, none of the aforementioned approaches consider the camera scheduling problem from a biometrics task-driven viewpoint. We propose to design a capturing quality-based objective function, which measures the overall success probabilities of all captures by all PTZ cameras, and search for the optimal scheduling strategy that decides the future behaviors of all cameras in a collaborative manner. Details of our technical approach can be found in Appendix D.

6.1.3 Experiments

The proposed algorithm was tested at the GE Global Research testbed with four fixed and four PTZ cameras. The layout of the testbed and the cameras can be seen in Figure 34 with example views shown in Figure 33).

We first present the performance of the quality metrics, where the technical details are described in Appendix D.2.1. Figure 31 shows the quality metrics (left) and PTZ parameters (right) for a target that walks along a straight line in positive X direction, passing underneath PTZ camera 2 (see Figure 34). One can see that, as the subject approaches the camera, the quality due to the decreasing camera-target distance improves, while the quality due to the view angle (increasingly down-looking) decreases. Furthermore, the physical angle limits of the camera and the trackability region, leading to a steep drop in quality, as the subject passes under the camera. The optimal capture region is at about $10\mathrm{m}$ in front of the camera.

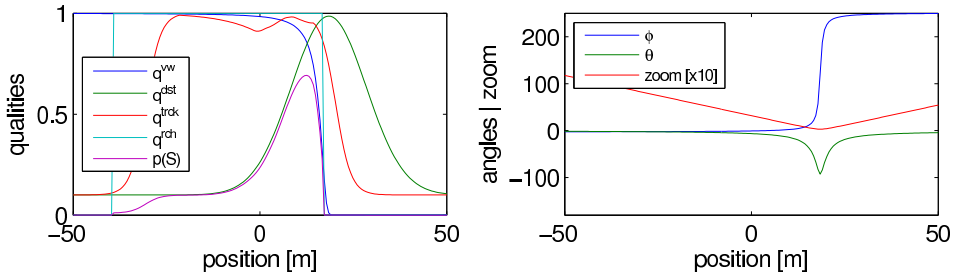


Figure 31: Quality Objectives and PTZ Angles.

Figure 32 shows several PTZ snapshots of a 7 minute video sequence. One can see from the Figure that the system (i) effectively capture all subjects in the field of view and (ii) prefers to capture frontal views of subjects. The system attempts to capture biometrically beneficial frontal views of subjects whenever it can, but will resolve to capture subjects from the side or behind, when given no other option. As an example of the behavior of the proposed algorithm in one instance in Figure 32 (bottom, left) the system mostly focuses on the subject in white shirt, being the only one facing any of the cameras. As the subject turns around and walks away from the PTZ, the system changes its attention on two other subjects that are now facing the cameras (bottom, right).

For another illustration on how the system schedules PTZ cameras intelligently, we examine the situations where two subjects are being tracked by the cameras and a third subject enters the site from the left (in terms of the Figure 34, top). The system only has very limited time to capture a frontal view and quickly re-targets PTZ 4 to follow the new arrival. Figure 33 (left) shows the success probability for all three targets. The third target (id=16) that was captured only from a side angle has a success probability of only $P(S|id = 16) = 0.18$.

We now explore the numerical performance of the proposed approach. Figure 34 shows one of the four fixed cameras of the surveillance system tracking several subjects. In the figure, the targets had just been acquired and each was captured several times by the four PTZ cameras. Upon execution of the plan optimization, the exhaustive search examined a total of 42592 plan candidates using 8.40 seconds. The best plan was determined to be $\mathcal{E}^* = [[24, 29], [28, 26, 26], [28, 28], [28, 28]]$.

Figure 32: Control of PTZ Cameras. Snapshots of the PTZ views, four views per snapshot.

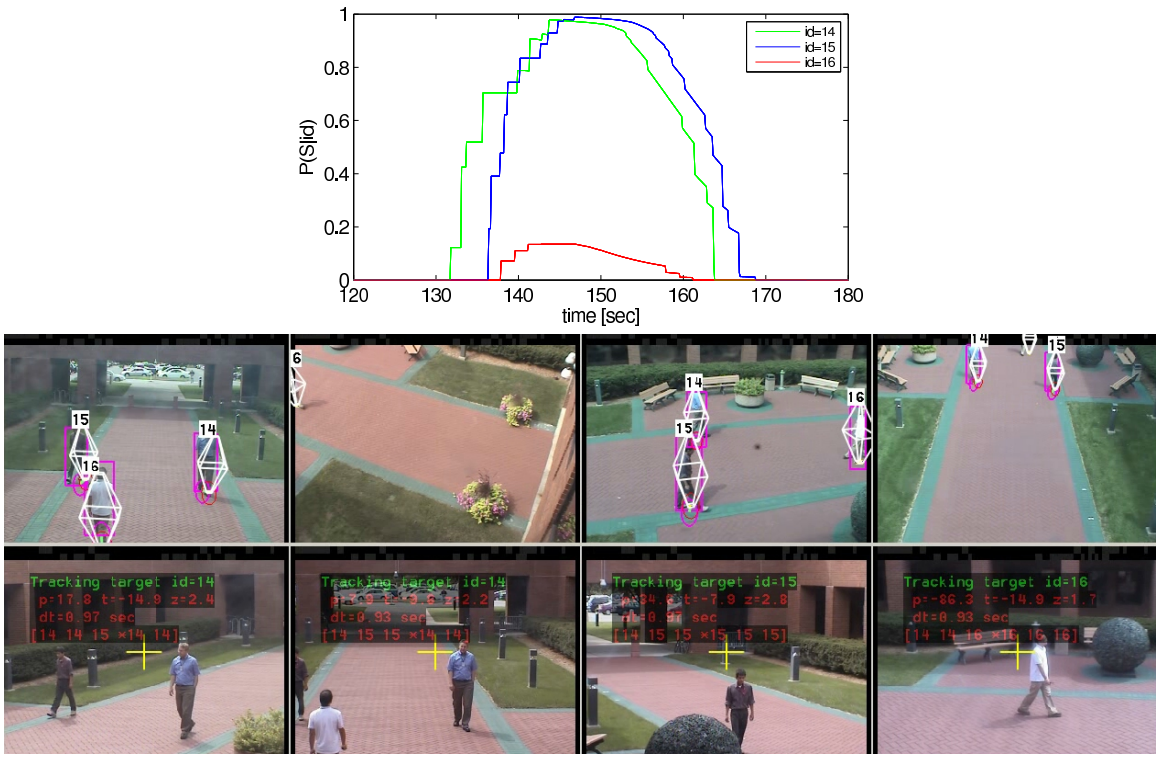


Figure 33: Limited Time to Capture Target. The top graph shows the target success probabilities. The bottom image shows the three subjects moving in the site. See text for details.

This plan was chosen, because target $id = 28$ had just been recently acquired by the tracker. The total plan probability was $p(S|\mathcal{E}^*) = 0.247747$. During optimization the best first search terminal node was found after visiting 100 nodes, which yielded the plan $[[29, 29], [26, 26, 26], [28, 28], [28, 28]]$ with plan probability $p = 0.235926$. The above optimal plan $\mathcal{E}^*$ was found after three steps of local plan refinements (see Table 1).

Table 2 for several plan optimization runs how the best first refined estimate compares to the global estimate and when it was reached (measured by the number of nodes that have been explored). This table shows that (i) the quality of the best first estimate is very similar to the globally optimal solution found through exhaustive search, (ii) the best first node is found early during the optimization process and (iii) very often the global optimum is reached after only a fraction of all possible plans. However, as shown by the extreme case in the first row of Table 2, the exhaustive search optimizer sometimes has to explore 50% or more of all nodes. In practice, if the plan horizon is not too large, the best first refined node is thus a very good solution to the control problem outlined in this work.

Step	Ref. Step	Plan [PTZ1, PTZ2, PTZ3, PTZ4]	Probability
100		$[[29, 29], [26, 26, 26], [28, 28], [28, 28]]$	0.235926
	1	$[[\mathbf{24}, 29], [26, 26, 26], [28, 28], [28, 28]]$	0.246558
	2	$[[24, 29], [\mathbf{29}, 26, 26], [28, 28], [28, 28]]$	0.247045
	3	$[[24, 29], [\mathbf{28}, 26, 26], [28, 28], [28, 28]]$	0.247747

Table 1: Plan Improvement during Optimization. The best first search followed by local label switching yields the optimal solution after 100 steps, while the exhaustive search finds it using 42592 steps.

The presented system was integrated on a dual CPU (Intel Quad Core) workstation running in real-time at around 15 FPS under load, processing four views at a resolution of 320×240 and capturing a total of eight views (no image processing was performed on the PTZ views).

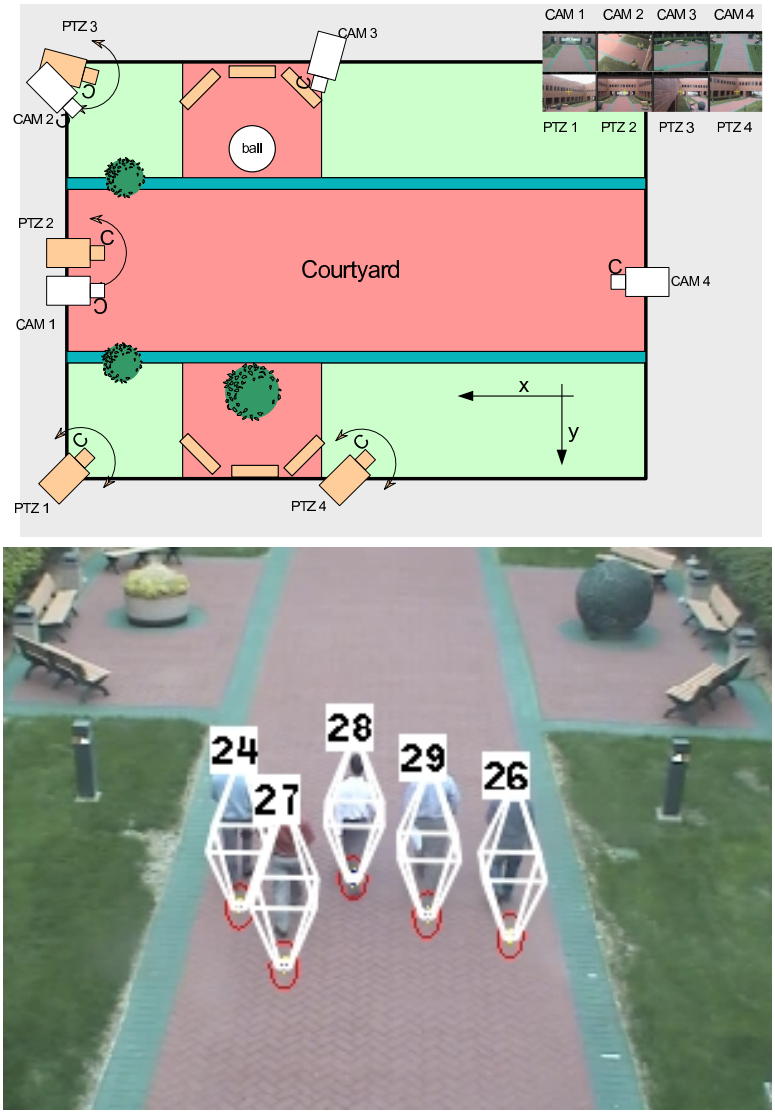


Figure 34: Multi-Camera Surveillance System. Top: The experimentation setup. Bottom: The employed tracker performs multi-camera multi-target tracking in a calibrated camera environment. The above images show five individuals being tracked by the system. The system performs centralized tracking, i.e., the identities of targets are maintained across camera views.

All Mock Prison Riot data was *enacted* by law enforcement and corrections practitioners.

total nodes	best first found at	best first probability	best node found at	best node probability
234256	81	0.148558	127866	0.161289
38016	72	0.191554	6407	0.201699
96668	112	0.267768	112	0.267768
36864	81	0.231526	8159	0.234387

Table 2: Best First vs. Best Node. The table shows the probabilities of the best first search result vs. the global optimum. It also shows when the best first node is reached and when the best node was found during the search. The left most column indicates the total number of nodes examined.

6.1.4 Discussions

We presented a systematic approach to achieving real-time collaborative control of multiple PTZ cameras. The goal is to optimize the captures of subjects in a multi-camera surveillance system with respect to a set of quality metrics, which depend on capture distance, view angle, target reachability and trackability. The reported experiments demonstrate the effectiveness and intelligence of the system to obtaining high-quality imageries of all site-wide activities. For active camera systems, detailed quantitative performance evaluation is challenging, since experiments can only be conducted in real-time.

6.2 Identity Maintenance

As a next step we added face detection and face recognition back-ends to the PTZ captures. The face detection system is responsible for capturing people's faces whereas the face recognition system creates a record of individuals that the system is seeing over time. We are utilizing a face recognition system from Identix to perform this task. Figure 35 shows a screen shot of the back-ends operating on the PTZ camera views. Table 3 shows the performance of the facial biometrics back-ends. In 2156 frames we detected 411 frontal faces, from which we made 353 correct (rank 1) recognitions.

A major challenge with identity maintenance as performed here is that face detections and face recognitions are obtained from the high-resolution, close-up PTZ camera views whereas the grouping and location information is obtained with the tracking system, estimated based on the

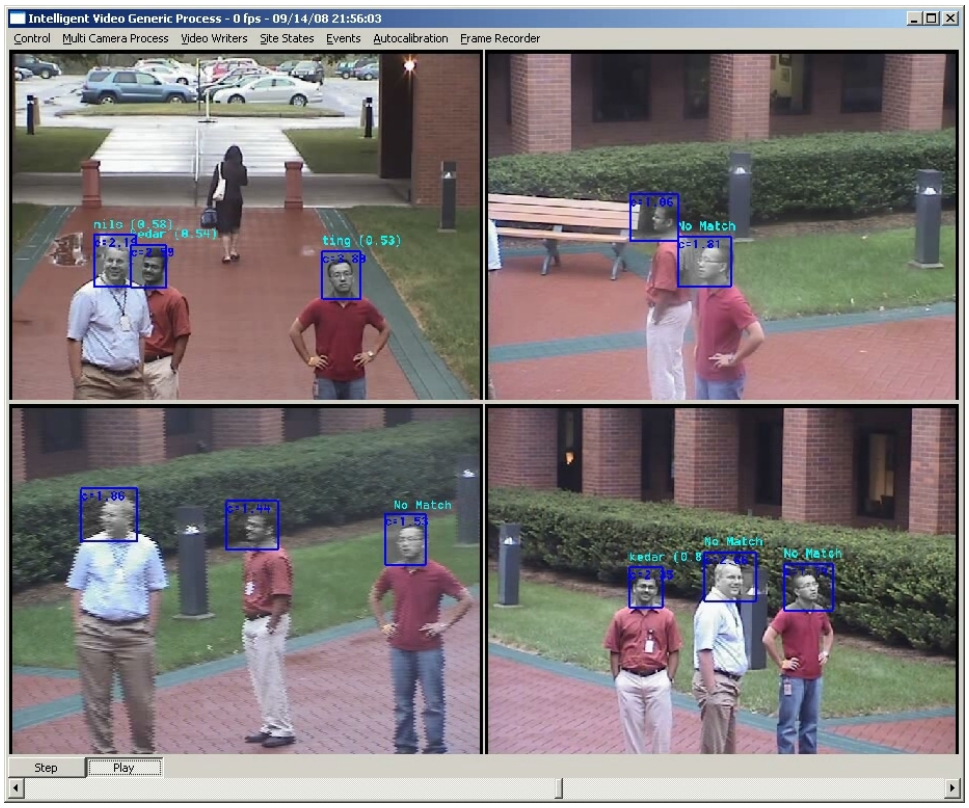


Figure 35: Face Detection and Face Recognition. Snapshot of the system performing face detection and face recognition. The blue rectangles denote locations where faces were detected in the PTZ views. Above the rectangles, names as estimated by the face recognition are shown. When "No Match" is shown, the face recognition could not determine a reliable identity estimate. When no name is shown, the face recognition system determined that no frontal face was present.

Nr. of Frames:	2156	Face Detections:	843
Frontal Detections:	411	Recognitions:	403
Correct Recog.:	353	Recog. Rate:	88%
Rank 2:	14	Rank 7:	1
Rank 3:	2	Rank 8:	1
Rank 4:	2	Rank 9:	1
Rank 5:	4	Rank 10:	1
Rank 6:	1	Rank $\geq$ 11:	23

Table 3: Face Detection and Recognition Performance: A small representative video segment was groundtruthed and evaluated for accuracy.

fixed camera views. This estimation of data in different camera views makes it necessary to perform data association between the faces in the PTZ views and the tracks in the fixed views. This task is challenging due to noise in the tracker as well PTZ control system. Figure 36 visually shows the task to be solved. In the PTZ views face detections and tracks are not uniquely aligned in the PTZ views. Furthermore, the PTZ view to world projections of face detections into the ground-plane coordinate system (wireframe ellipsoids in the zoomed-in views in Figure 36) are inherently ambiguous.

In the next section we presented a sophisticated method for solving this problem, allowing us to associate names, obtained with the face recognition system, with tracks, obtained by the tracker. Figure 37 schematically shows the problem solved by our system and Figure 38 shows the result, after association.

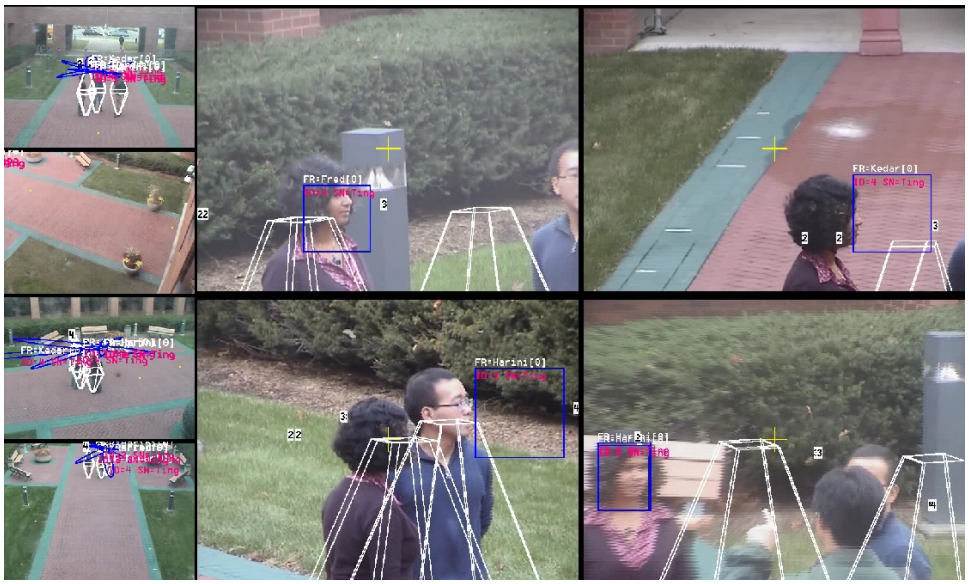


Figure 36: Face to Track Association. This figure shows the track views (four small images on the left) and the PTZ views (four large images). One can see the noise that makes the face to track association challenging (see text).

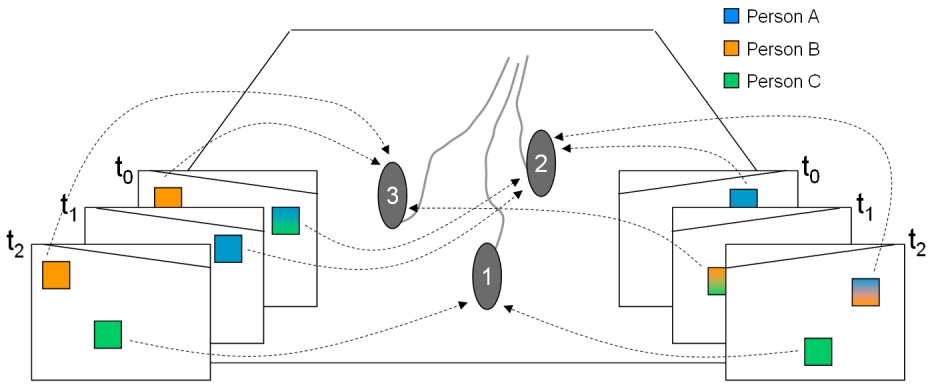


Figure 37: Face to Track Association Challenge. Schematic problem that the identity maintenance has to perform. Face recognitions (frames at t_0, t_1, t_2) of subjects A,B, and C have to be assigned to tracks 1,2, and 3. In addition to noisy localization (see Figure 62) the recognition might give ambiguous results (e.g., in right frame at time t_2 the recognition gave a 50/50 chance to the person detected being A or B).

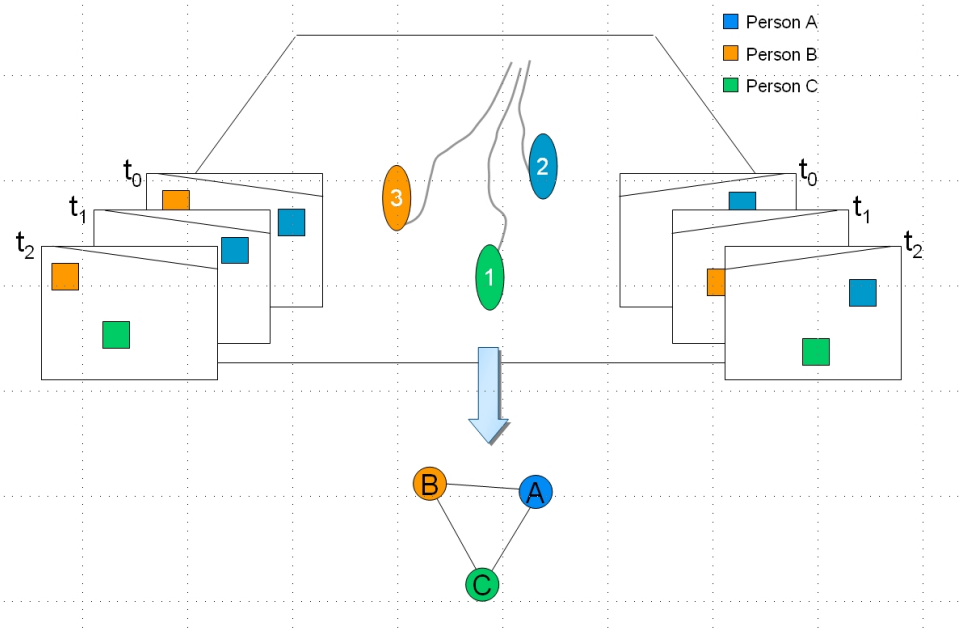


Figure 38: After Association. The data association has processed all tracks and all detections and made a final name to track assignment. The recognition in the right frame at time t_2 was corrected to be person A. The interaction graph associated with this scene is shown on the bottom of this figure.

7 Social Network Estimation

Having the ability to associate identity information with target tracks now allows us to measure the degree of social interaction between people.

This section is adapted from a self-contained technical paper entitled "Monitoring, Recognizing and Discovering Social Networks" that was published at the IEEE International Conference on Computer Vision and Pattern Recognition by Ting Yu, Sernam Lim, Kedar Patwardhan, and Nils Krahnstoever.

7.1 Introduction

The goal is to gain a higher level understanding of crowd behavior in terms of interaction and social network patterns. A *social network* consists of groups of people with a pattern of interactions between them [22] and the understanding of such networks in environments such as prisons or public venues is of great interest to law enforcement and homeland security. In particular, there is an increasing need to identify cohesive groups, which we called *social groups*, and their leaders for security purposes. One can easily imagine, for example in a prison environment, the value of automatically identifying different *gangs* and their *leaders* as well as changes over time in gang and leadership structures.

Here, the low-level vision tasks that need to be performed *reliably* include

1. Persistently tracking an individual under occlusions. These tracks allow the system to detect individuals that are often "seen" together and assign them to the same social group. For a multi-camera setup, centralized tracking in a common coordinate system is required.

2. Uniquely recognizing an individual on a watchlist using face detection and recognition [23, 24]. For this purpose, high-resolution images of faces are required.

Towards this end, advances in real-time tracking [25, 26, 27] and PTZ camera control algorithms [28], that are capable of effectively panning, tilting and zooming PTZ cameras to capture

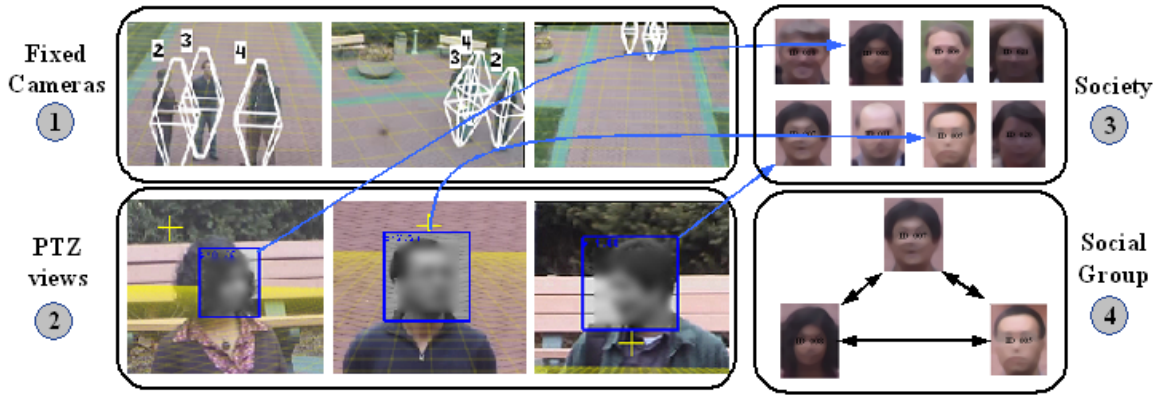


Figure 39: Discovering Social Networks: Person tracks obtained from fixed camera views are used to control the PTZ cameras to capture face images. Faces recognized from these images and person tracks are in turn used to build the social network or society. The social interactions captured by the social network form the basis for discovering the social groups.

high-resolution face images of people, have made it possible to perform these low-level tasks in an efficient and robust fashion.

Under this premise, we assume that the 3D tracks of individuals and high-resolution face images are provided, allowing us to focus on the high-level task of analyzing and discovering social groups in a social network. As far as the authors are aware of, this is a new problem in the computer vision community. Arguably, previous work on group tracking [29, 30, 31] bears some resemblance to social grouping, but a few important differences will become clearer as we continue.

To begin with, the identities of individuals have to be maintained with respect to a (possibly dynamically generated) watchlist of faces. That is, during tracking, individuals are identified by performing face recognition. This allows the system to reason about interactions between different individuals, e.g., Joe and Frank have been together for the last five minutes. In this way, connections between individuals, represented in a social network graph, who are frequently seen together, become progressively stronger over time. The nodes in a social network graph represent the individuals as identified by their faces and the edges are weighted according to the observed connections between nodes. Such a social network graph can be built over time, even offline, as

long as individual tracks and captured face images are stored. When a social network graph is divided into social groups, each group conveys certain high-level information (e.g., Joe is the leader of the group comprising Frank, John and Tracy), which is not possible with mere group tracking.

Building such a social network graph and discovering its social groups correctly is non-trivial. Several challenges involved in this process have to be tackled and we propose robust solutions to address these challenges in this work. Noisy tracking and face recognition typically "corrupt" the social network graph, which could potentially mislead the division of the social network graph into incorrect social groups. Moreover, given a set of tracks (observed from a fixed camera network) and recognized faces (observed from a bank of PTZ cameras), one has to associate each face with its track correctly, a difficult task due to errors in calibration, uncertainty about the PTZ camera state, and the proximity between tracks under crowded conditions. We overcome these challenges with an energy function that elegantly models the association problem and minimize it using the multi-way graph-cut algorithm [32, 33, 34].

The captured social interactions form the basis for dividing the social network into social groups. Essentially, we are faced with a graph based clustering problem, for which most previous work such as [35, 2, 36, 37] have focused on a variety of techniques for minimizing the size of the cut, which is defined as the strength of the edges that connect two disjoint clusters. We introduce a technique called the modularity-cut, originally proposed by Newman [22, 38] in the domain of social network analysis.

The modularity-cut proposed here is an appealing technique for discovering social groups because of its computation of group assignments in a way that is very meaningful in the "social" sense. The basic idea is that a cut should not be determined merely by its size, but rather by its size relative to the expected size (i.e., when the connections between groups are *smaller than expected*). As we will see, this is a powerful idea, as it inherently does not need to know the size of the social group beforehand (most cut based techniques require some way of checking for the trivial case where all nodes tend to fall into one group, in which case we obtain a zero cut size). Moreover, it provides for a simple way to discover the number of social groups and their leaders automatically.

The contributions of this work can be summarized as follows:

1. We address the problem of discovering social groups in, possibly closed-world, surveillance environments, which is an emerging challenge for the computer vision community. The goal of identifying cohesive groups and their leaders has tremendous value for practical surveillance purpose.

2. We present an elegant energy function to model the problem of associating tracks (in fixed camera views) to face detections (in PTZ views). This will allow us to construct a social network graph while mitigating errors coming from low-level vision tasks.

3. We introduce the modularity-cut algorithm that (as far as the authors are aware of) has not been applied to the domain of computer vision. The modularity-cut has several inherent advantages over classical clustering techniques. The algorithm also leads to an elegant way of estimating the group leaders. This is an important difference from mere group tracking.

4. We have built a fully integrated system that produces very good results, as described in the experiments (Section 7.2). The system consists of a centralized 3D tracker that utilizes multiple fixed cameras. The tracker performs very robustly under occlusions, and effectively controls a set of PTZ cameras to capture close-up face images. The tracks and the captured face images serve as input for building the social network graph and discovering social groups.

Technical details of how we construct the social network graph, associate faces with tracks, analyze a social network graph using modularity-cut, and identify the group leaders via modularity based Eigen-analysis can be found in Appendix E.

7.2 Experiments

In this section, we present results from a fully integrated system that consists of 4 fixed and 4 PTZ surveillance camera, capturing a total of 8 views. The fixed cameras are utilized by a centralized

3D tracker that provides the 3D locations of individuals in a common coordinate system. As shown in Figure 42 bottom image, the tracking capabilities of our system under occlusions are very good, being able to track a dense crowd of individuals in a small courtyard with relatively small number of errors. The tracks are then used to control the PTZ cameras to capture face images at high resolution. Referring to Figure 39, the system performs face detection and recognition on these face images, and the recognized faces are then associated with the tracks to build the social network graph. The steps for building the social network and discovering social groups can be optionally performed offline as long as the tracks and face images captured online are properly stored. For the remainder of this section, we will present results from various aspects of the system, namely face detection and recognition, face-to-track association, discovering social networks and identifying group leaders.

Face Detection and Recognition: We firstly establish the performance of the face detection and recognition components, which we evaluated on a short section of video, containing only single subject. This allows us to easily gauge the recognition performance, obtaining the results in Table 4. In this case, the system manages to capture faces in about 40% of the frames out of which 49% are deemed to be high-quality frontal faces. Over 98% of these are recognized with a correct recognition rate of about 88%. In our experiments, we have also observed that the recognition confidence (the score returned by the recognition engine) for correct matches is significantly larger than the confidence for incorrect matches.

Nr. of Frames:	2156	Face Detections:	843
Frontal Detections:	411	Recognitions:	403
Correct Recog.:	353	Recog. Rate:	88%
Rank 2:	14	Rank 7:	1
Rank 3:	2	Rank 8:	1
Rank 4:	2	Rank 9:	1
Rank 5:	4	Rank 10:	1
Rank 6:	1	Rank ≥ 11:	23

Table 4: Face Detection and Recognition Performance: A small representative video segment was groundtruthed and evaluated for accuracy.

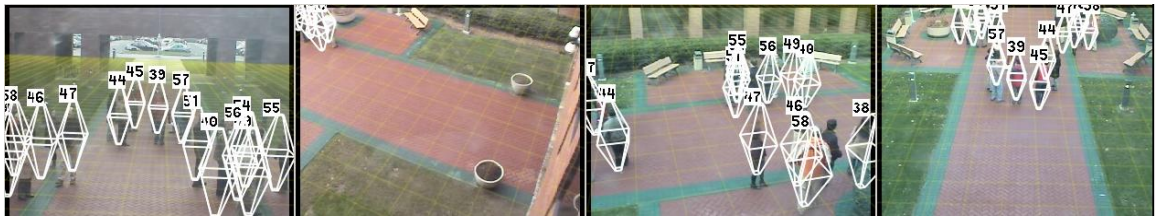


Figure 40: Large motion crowds: In this scene, there was a total of 23 individual subjects who were told to mingle in a 3-group configuration. The overlaid white wireframes with IDs show the tracking result from 4 fixed camera views.

Face-to-Track Association: To evaluate the performance of our proposed association algorithm presented in Appendix E.1.1, the ideal experiment would be to compare the face-to-track associations returned by the graph-cut solution to the groundtruth obtained from manually associating faces with their tracks. This is, however, a prohibitive task, considering the large number of recognized faces even for a single track, as seen in Table 4. Instead, we overcome this by manually labeling each track (the number of tracks is significantly smaller than the number of faces) with the identity of the individual that this track is following. On the other hand, given that multiple faces are associated with each track in our graph-cut solution, we perform majority voting whereby the most frequently recognized individual for this track is assigned to it. Furthermore, for computational reasons, our graph-cut optimization is performed for temporally partitioned segments of the tracks. Therefore, the majority voting procedure is conducted for these track segments. The groundtruth labels and the labels from our solution are then compared.

	Seq #1	Seq #2	Seq #3
Nr. of Frames:	7000	5400	7000
Nr. of Tracks:	20	14	22
Nr. of Segments:	364	267	904
Recognized:	352	264	597
Correct Recog.:	336	255	470
Wrong Recog.:	16	9	127
Recog. Rate:	95%	97%	79%

Table 5: Track Recognition Performance: We groundtruthed 3 video sequences by labeling each track with the identity of the individual that it was following, and compare them with the the results from the graph-cut solution, see text for details.

Following such a procedure, we groundtruthed 3 video sequences that contain a total of 19400 frames and 56 tracks. We show the comparative results in Table 5. Seq #1 contains 20 tracks, generating a total of 364 segments, among which 352 segments are recognized (some segments may not have been associated with any faces, and thus remain unrecognizable). From these 352 segments, 336 are correctly recognized while only 16 are wrong, yielding a recognition rate of 95%, which is higher than the face recognition rate in Table 4. For Seq #2 and Seq #3, we obtained a recognition rate of 97% and 79% respectively. While the recognition rate for Seq #3 is lower than the face recognition rate in Table 4, the amount of uncertainties for the latter is significantly lower since the test sequence contains only a single subject. On the other hand, we are faced with uncertainties caused by crowded conditions, errors in the projection matrix estimations of the PTZ cameras, and motion blur due to PTZ movements. Thus, we consider the overall performance of the 3 sequences very satisfactory.

Discovering Social Networks: We have evaluated the robustness of our system in discovering social network extensively, and show here a typical scenario in Figure 40. A total of 23 human subjects were invited to participate in our experiments. Our system managed to track each individual quite reliably even under such a challenging condition, as seen from the 4 fixed camera views in the Figure. The participants were instructed to mingle in a 3-group configuration. Given such a typical scenario, a social network was estimated and shown in Figure 42 left image. Based on the social network graph, we proceeded to discover its social groups. The modularity-cut was able to discover the correct social groups, shown in Figure 42 right image.

We have also compared the modularity-cut with recursive division based on the normalized-cut [2] criterion. The normalized-cut was not able to generate the correct groups when applied to the social network graph in Figure 42, misplacing 1 individual in another group due to the weak connections that individual has with all the groups as a result of noise in the dataset.

Moreover, by analyzing social network graphs containing different number of social groups, we identified two main problems when using normalized-cut. Firstly, it is non-trivial to set the cut size threshold, and we often have to attempt several different threshold values to achieve good

performance. Secondly, it seemed that the normalized-cut is biased in favor of equal-sized social groups, which causes problems when the social groups are very uneven in group size. In contrast, modularity-cut was able to correctly discover the social groups in most cases.

Leadership Identification: To demonstrate our algorithm's capability to identify Eigen-leaders, we run the modularity-cut on a long sequence, whereby different members of two groups are seen interacting at different times. The leader of each group is, however, always present, which generate strong modularity connections between each leader and his group members. By identifying the resulting Eigen-leaders, the system was able to successfully identify the leaders, shown as red nodes in Figure 41, which, as mentioned, is not possible with classical spectral clustering techniques that minimize the cut size.

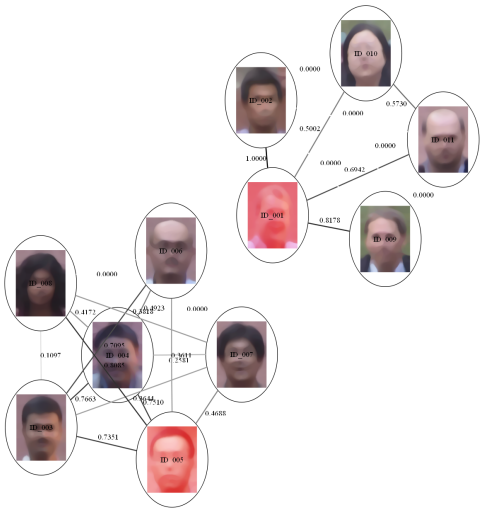


Figure 41: Leaders of Social Groups: In this experiment, there was a total of 11 individuals, divided into 2 groups. Members of each group appears in the scene at different times, during which the leader is always present. This generated, in a modularity sense, strong connections between the leader of each group to its members. As a result, the system was able to identify the Eigen-leaders (Appendix E.2.3), which are the red nodes in this figure.

7.3 Conclusions

We have addressed an emerging new problem of monitoring and recognizing social network and discovering its community and leadership structures. A computer vision solution to the prob-

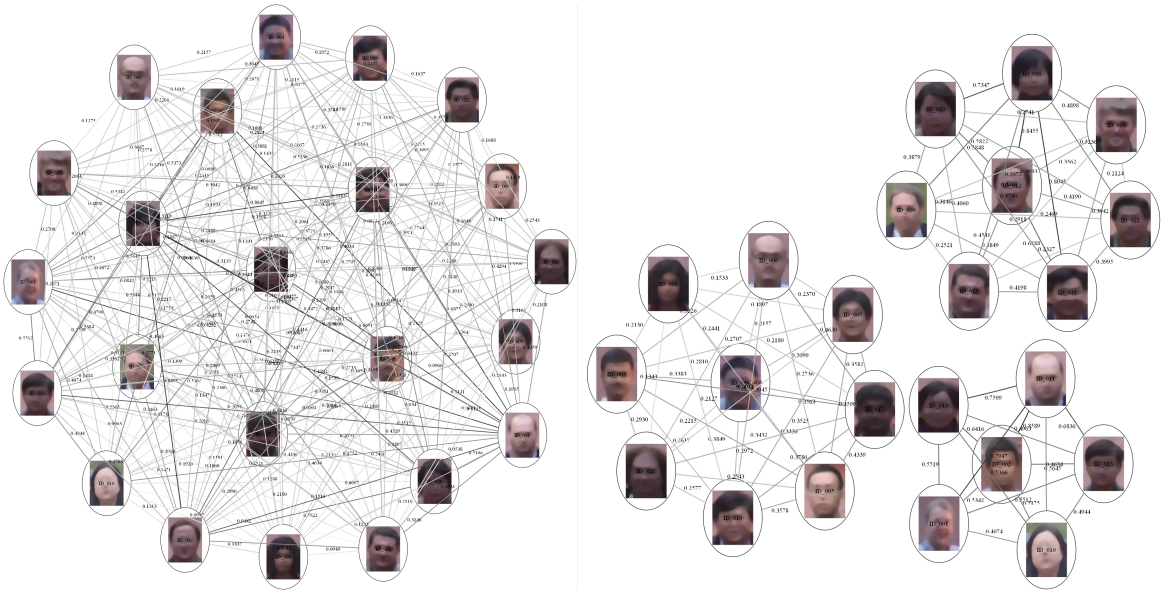


Figure 42: Discovering Social Groups: The left image shows the social network constructed from the sequence in Figure 40, which was divided into 3 groups correctly using the modularity-cut, shown on the right. Here, the links are shown with the weights between the individuals.

lem has been proposed, which should be of tremendous value for practical surveillance purposes. Specifically, we have presented an algorithm for robustly building social network by identifying individuals and analyzing their interactions using face recognitions and person tracks, where the correspondence between them are established using an elegant graph-cut solution. Admittedly, analyzing social interactions using such information alone might be limited. A more sophisticated scheme, such as the amount of eye contacts, could be employed. These schemes, however, are mostly very sensitive to video quality and thus require further work. We have also introduced the modularity-cut as a more intuitive way of dividing the social network into social groups, and demonstrated its capability in identifying Eigen-leaders. The utility of our algorithms was demonstrated by a fully integrated system that works very well in practice; the system can be run live for an extended period of time collecting sizeable amount of tracks and face images for subsequent social network analysis. The system in its current form, however, assumes a closed-world environment where the watchlist is pre-determined. This should be desirably extended to a dynamically-generated watchlist, where new faces observed during system operation are added to the watchlist

automatically.

We also experimented with non-face recognition cues. An experiment to utilize gaze is shown in Figure 43. This cue is in practice not very reliable as it is difficult to measure and sometimes misleading. Two interacting people often don't look at each other, which is especially the case for people that know each other well. Gaze hence appears to have limited power to express social interaction.

Figure 43: Gaze for Social Interactions. Gaze can be utilized to get a measure of the interaction between people in a group. In practise, this cue is challenging to utilize because people that interact might not always look at each other. The above scene involves two people talking with each other while walking.

8 Data Collection and System Testing at Mock Prison Riot 2009

8.1 Collection and Testing Approach

As described previously in Section 4.3, we conducted a data collection and testing phase during the 2009 Mock Prison Riot at the decommissioned West Virginia Penitentiary in Moundsville, WV.

The goal of the Mock Prison Riot was to (i) collect video data of law enforcement relevant activities in a correctional environment and (ii) test the current system as developed by GE Global Research under NIJ funding. The approach to data collection was the following: several law enforcement teams that participated in the Mock Prison Riot volunteered to enact relevant correctional scenarios. Unlike the official Mock Prison Riot scenarios, which focused on law enforcement *reacting to* disorderly conduct, the enacted scenarios focused on the activities and behaviors *leading up to* the disorderly conduct. In other words, the GRC data collection was aimed at capturing the events that are leading up to fights between inmates, assaults etc., rather than how law enforcement handles the after-math of such developments. The GE team gave only basic instructions to the volunteers and asked for enactments of scenarios that LEC officers would like a system to detect automatically. Teams quickly started to enact a variety of scenarios that had never even be considered during the initial system development.

The teams that participated in the research specific data collection where the following:

- Criminal justice students from the University of Utah,
- Lake Erie Correctional Institution,
- Montgomery County Correctional Facility,
- Delaware County Sheriffs Office, and
- Chatham County Sheriff's Department.

In addition the official MPR scenarios where recorded, which in addition included teams from:

- North Dakota State Penitentiary,
- USP Hazelton,

All Mock Prison Riot data was *enacted* by law enforcement and corrections practitioners.

- Hudson County Dept. of Corrections,
- Naval Consolidated Brig Miramar,
- Federal Bureau of Prisons, and
- Lubbock County.

8.2 IRB Approval

The experimental data collection protocol and consent procedure have been approved by an IRB Board (IRC). All volunteers that enacted scenarios specifically for the NIJ research program where informed about the purpose of the study, that participation is voluntary, that anyone was able to ask to discontinue participation at any time and that no identifiable information would be shared with outside parties. All volunteers signed informed consent forms before data collection.

8.3 Collected Video Data

Appendix C provides an overview of the main sequences that where captured during the Mock Prison Riot. The sequences are listed in three separate tables. The first table lists sequences captured by the processing system which was stopped and started more frequently and labeled in more detail. The second table lists all scenarios that we have examined in more detail as part of this program and the third table lists the data that was captured by the recording system which was typically set to record continuously during events and scenarios and is labeled in less detail. Table 6 below contains a summary of the total amount of data recorded. The table is broken down into the duration of all relevant video sequences (which exclude system tests, calibration etc) and total amount of video sequences. In total about 23.7 hours of video material was captured during the four days of the Mock Prison Riot event. The number of frames recorded has to be multiplied by the number of deployed cameras (i.e., four) to obtain the total amount of footage. Overall 7139656 video frames (more than seven million) were captured. The stored size of the footage is exceeding $120GB$.

Recording Type	Length [Frames]	Length [hours]
Processing Machine (all)	825045 x 4	14.1
Processing Machine (relevant)	276000 x 4	5.7
Recording Machine (all)	959869 x 4	9.6
Recording Machine (relevant)	675986 x 4	6.7

Table 6: Amount of Recorded Data. Number of frames and total duration of video recordings performed on the processing machine and the recording machine.

8.4 Mock Prison Riot Detection and Tracking

The GE Global Research tracking system performed well under the scripted conditions during the Mock Prison Riot even though the environmental conditions where at times challenging, varying between heavy rain, clouds and bright sunshine (Figure 44).

During events that have not been organized by GE Global Research, flash bangs and tear gas simulators (smoke) where often utilized, which made video tracking difficult. In addition during some scenarios that involved a large number of officers "piling" on disorderly prisoners, tracking individuals also became impossible (Figure 45).

Figure 46 shows an official MPR scenario as an example, where the tracking system worked well in the beginning of the scenario, but then degraded in performance due to the use of smoke granades, shaking cameras and dense crowds.

8.5 PTZ Control

One of the cameras used for data collection was a Sony EVI-D70 that could be "slaved" by the tracking system to follow isolated individuals over time as per the algorithm described in Section 6.1 of this report. This camera was responsible for obtaining high-resolution facial captures of subjects during the scenarios and will in practice enable the social network estimation and maintenance.

All Mock Prison Riot data was *enacted* by law enforcement and corrections practitioners.

Figure 44: Tracking. The tracking system performed well even under crowded and active conditions: (a) actors playing dodge ball, (b) tracking in rain (notice the strong reflection of the person, which can sometimes lead to false person detections, (c) tracking during an actual scenario, and (d) tracking during a contraband exchange.

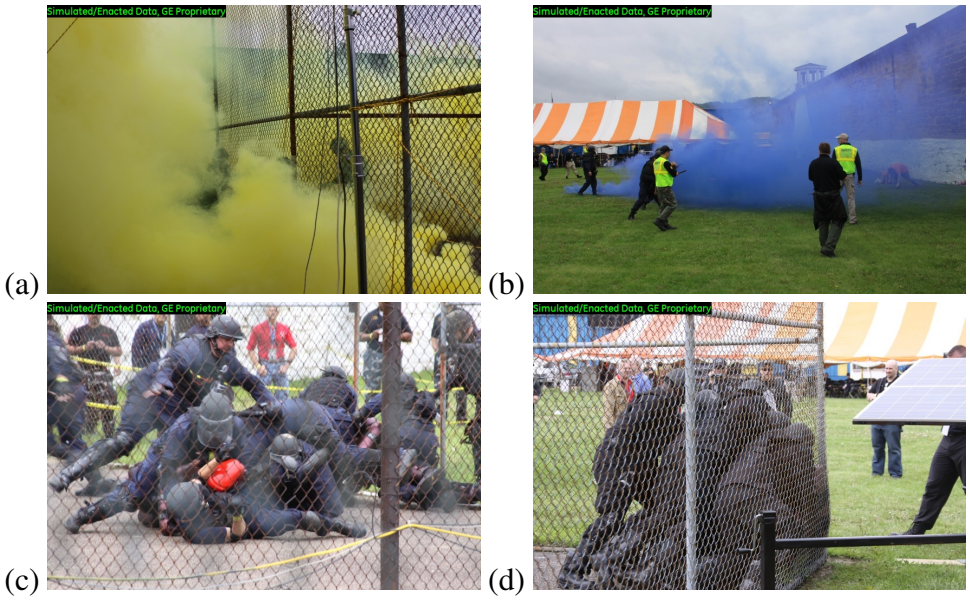


Figure 45: Extreme Conditions. Under these conditions the video system was no longer able to resolve individuals: (a) and (b) dense smoke, (c) and (d) dense fights.

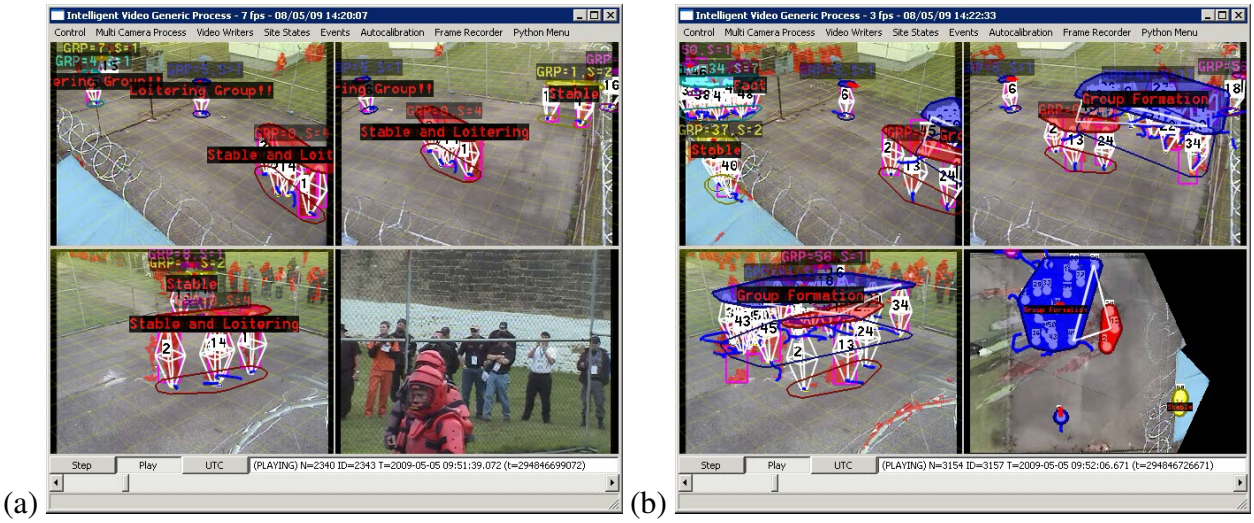


Figure 46: Extreme Conditions During Scenario. In the beginning (a) of MPR organized scenarios, when actors (here three officers in "Red Man" suits) where waiting for the arrival of the Special Response Units, the tracking system worked well. When the response units overwhelmed (b) the rioting inmates, and smoke grenades where used, the system degraded in performance with many false alarms being caused by smoke and shaking cameras.

Figure 47: PTZ Tracking. These captures show the PTZ control algorithm in action. This figure contains nine screen shot with a 2x2 arrangement of video frames. The PTZ camera (bottom right view in each of the 2x2 visualizations) is slaved automatically to the tracking system with the goal to acquire high-resolution facial shots of subjects. This sequence shows how the system automatically transitions between the three subjects.

8.6 Behavior and Event Recognition

The GE Global Research system is designed to recognize a variety of events and motion patterns from video.

- loitering prisoner / group
- fast moving subject
- counter moving subject
- following / chasing event
- group formation / dispersion
- approaching detection
- aggression detection [*]
- distinct group/gang detection [*]
- flanking detection [*]

The behaviors marked as [*] where not available during the live Mock Prison Riot operation but rather have been developed in response to the experience gained during the event.

8.6.1 Meeting / Approaching / Contraband Exchange

The process of people meeting or approaching one another can be an indicator for activities such as contraband exchange or a fight. Our system has the ability to detect such movements, which led to a range of salient event detections:

An example output of the system and detected event is shown in Figure 48. The figure shows the original screen output (a) as well as a capture of the scenario (b)-(d) that was detected. One can see in (a) that our system has a tendency to provide too much visual feedback to the operator. This will have to be improved in future versions of the system. The original screen output shows

three separate events: a group loitering, the actual counter flow described above and a second group loitering partially occluded by a member that just joined the group (flagged as "relatively fast"). The PTZ view shows just the top of the heads of the two individuals meeting exchanging contraband.

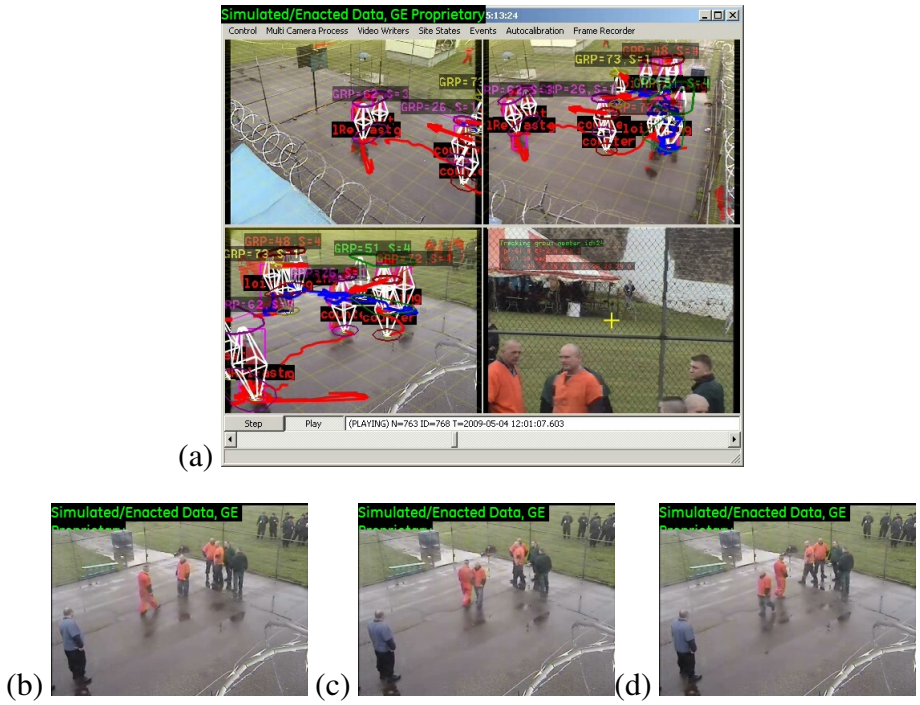


Figure 48: Contraband Exchange. Example of a detected contraband exchange, flagged by the system as a suspicious "counter" motion between two prisoners. (a) quad view that shows the original screen capture, (b)-(d) progression of the event.

Figure 49 shows a second example event flagged by the system, where two groups perform multiple contraband exchanges. One can again see the somewhat overloaded visualization by the system (a) and the detected event (b). Other meeting events can be seen in Figure 50.

8.6.2 Aggression Detection

Figure 51 shows a fight between two gangs where not the actual fight was detected but rather the "fleeing action" after the fight. At the time of the Mock Prison Riot, only the first agitation detection method described in Section 5.7 was available, which could not be executed during the live operation due to CPU and detection performance issues of that sub-component. We have since

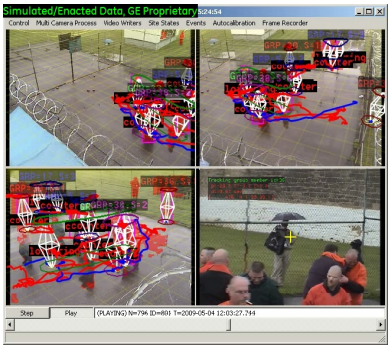
(a)  (b) 

Figure 49: Gang Contraband Exchange. Example of a detected contraband exchange between two gangs, flagged by the system as a suspicious "counter" motion between two groups.

(a)  (b)  (c) 

Figure 50: Meeting Event. Example detections of individuals "meeting". The (b) and (c) show that the actual meeting event is an indication for a fight between individuals.

improved the aggression detection system, the result of which can be seen in Figure 52. The new aggression detection is described in more detail in Section 5.8.

8.6.3 Fast Movement

Fast moving or running individuals are in general an indication for unusual events in a prison. The fast subject detection worked very well during the Mock Prison Riot and fast subjects where consistently detected by the system (see Figure 53).

8.6.4 Distinct Group Detection

After analyzing the Mock Prison Riot data in detail, it became apparent that an additional method for detecting separate, possibly opposing, groups is needed. The group analysis that was developed during previous program phases has the ability to perform a temporal clustering of individuals into groups, but the grouping by this approach tends to change over small temporal scales. A second

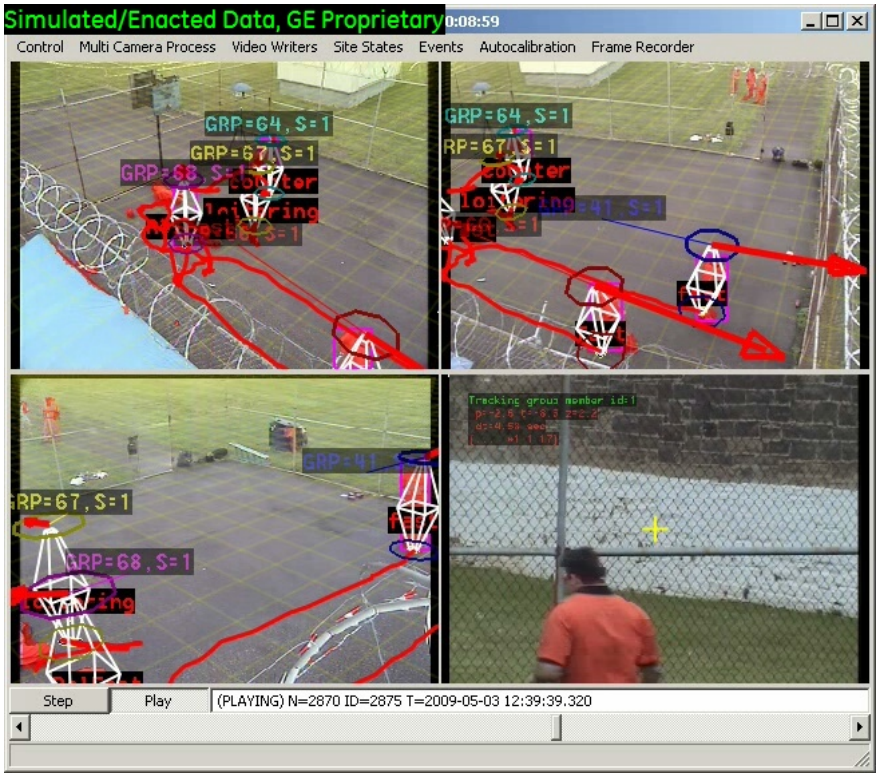


Figure 51: Fleeing after Fight. The live processing system did not run the aggression detection algorithm due to the computational requirements by that system. The system rather detected the action of inmates fleeing from a fight, flagged as "Fast" subjects.

Figure 52: Aggression Detection. Aggression detected by the new GE Global Research algorithm.

(a) 

(b) 

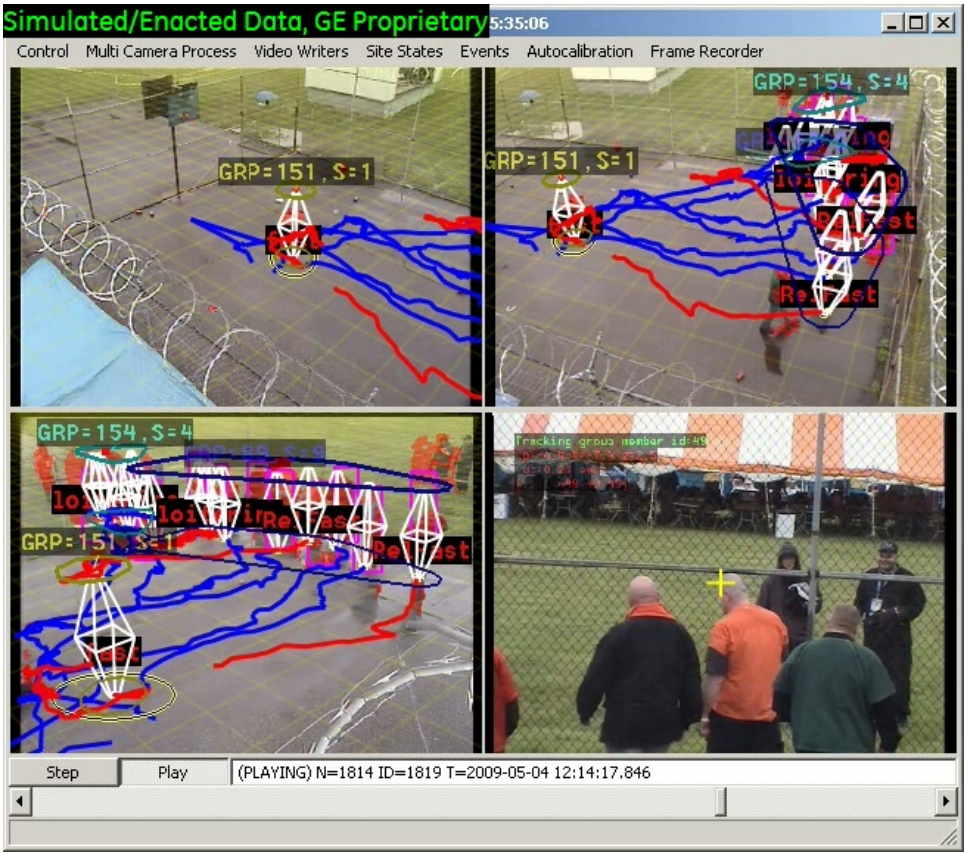
(c) 

Figure 53: Fast Motion Detection. Fast individuals detected by the system. These example views show an officer running to apprehend individuals that had assaulted a fellow inmate (a) and an officer running to investigate a detected contraband exchange (b). The bottom view (c) shows fast moving events detected after the members of an opposing "gang" stabbed an inmate.

layer of semantically higher reasoning was needed to maintain the following (see also Figure 54):

- Information about the *stability* of a group, i.e., a measure of how much (or rather how little) change a given group has experienced. A stable groups is viewed to be one that doesn't undergo changes in membership.
- Information about the *movement* of a group, i.e., a measure of it's spatial movement over time. The concept of *loitering* was extended to the group concept.

Figure 54 shows an advanced view of the MPR site where on the right hand side a virtual top-down view of the action is presented. The top-down view provides a better operational picture of the surveillance environment than the individual camera views.

Based on the above stability concepts a layout detector was developed that determines whether there exist a pair of stable groups of a minimal size that undergo little movement. Pairs of such groups can be viewed to be groups that are separate due to some higher-level reason such as *gang affiliation* or *community membership*. Figure 55 shows two example of such "distinct group" detections.

8.6.5 Flanking Detection

Several of the scenarios that were enacted by the volunteer officers contained the actions and activities that led up to different type of fights between gangs. Since our goal is to detect *predictors* of disorderly conduct we are particularly interested in the events leading up to a fight.

Analyzing several scenarios it became apparent that events leading up to a group-fight can be described as *two distinct groups approaching each other slowly with one or both groups spreading out in order to partially surround the other group*. In essence, groups are naturally trying to out-flank each other in order to attack from several directions simultaneously. We have hence adapted our event detection framework to incorporate *flanking* detection. The approach is based on analyzing the width and distance of pairs of groups as well as the degree to which one group is surrounding another (see Figure 56).

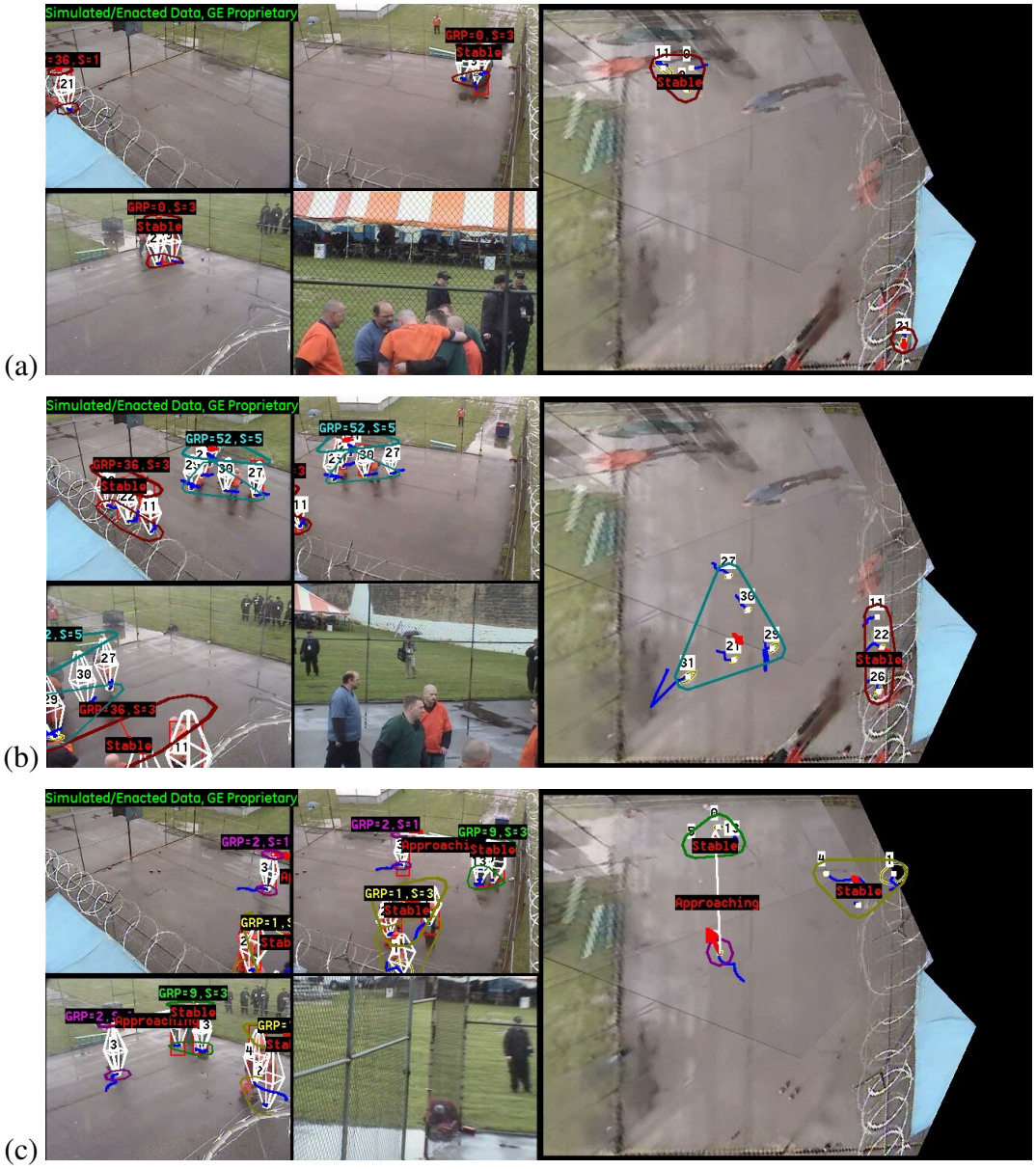


Figure 54: Stable Group Detection. A detection algorithm has been developed that detects groups that are stable over extended periods of time with respect to it's membership. In (a) the red group is in a huddle and determined to be stable. In (b) one of the groups is stable. The stable group is approached by a second group which changes more frequently. (c) Two groups are stable. One of the groups is approached by an individual.

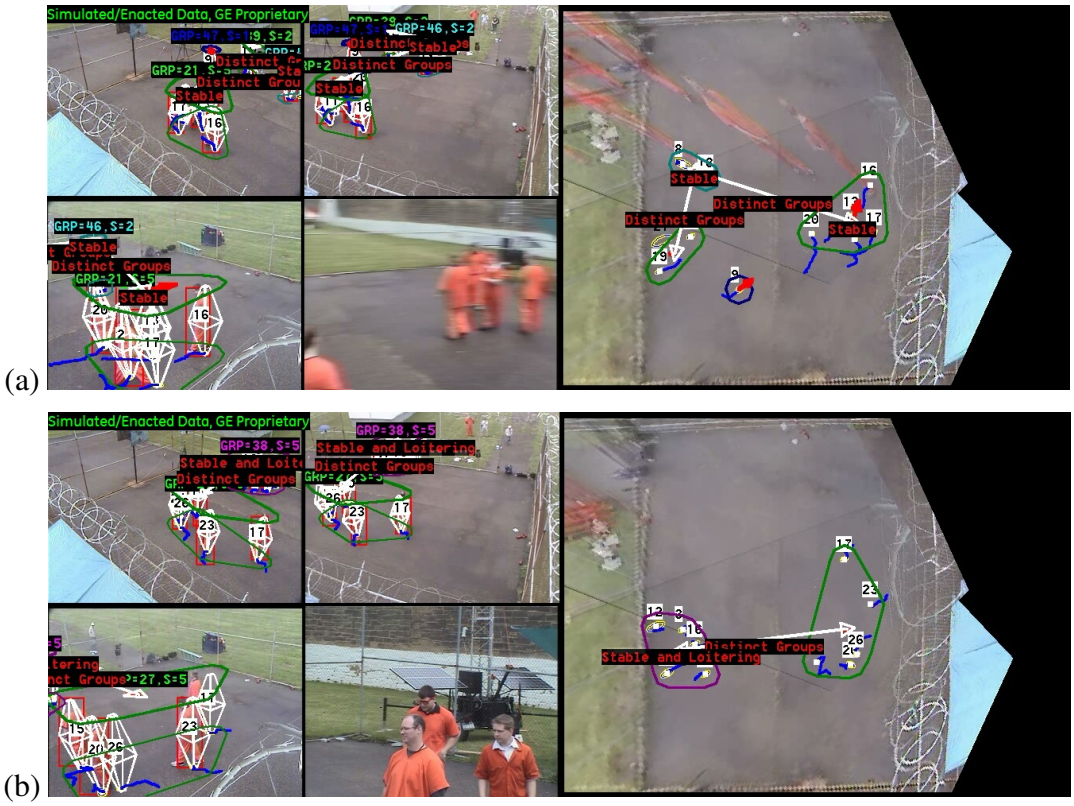


Figure 55: Distinct Group Detection. A detection algorithm has been developed that detects pairs of gangs that are stable and have certain spatial configurations with respect to one another. In (a) there are three groups, two pairs of which are considered mutually distinct. In (b) one group (red) is considered stable and loitering. It is considered to be distinct from the second group (green).

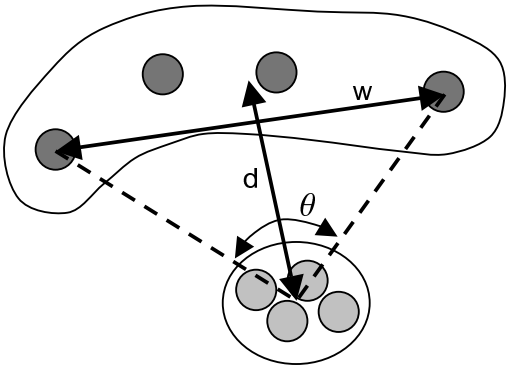


Figure 56: Flanking Detection Scheme. A group (top) is viewed to flank another group (bottom) if the angle θ exceeds a certain threshold and the distance between the groups d is smaller than the width of the flanking group w.

Figure 57 shows two of the correctly detected flanking events. In both instances the system detected the *onset* of a fight between two gangs, essentially fulfilling the vision of an automatic video system that has the ability to predict the occurrence of criminal activities in a law enforcement relevant setting. If a guard would be alerted by the system at that time, there would be a fair chance that the fight could be avoided either by rapidly deploying officers to the scene of the event or by doing intervention by audio as is often done in video monitoring where surveillance operators can let people know that they are being watched through video cameras (a very strong deterrent).

(a)

(b)

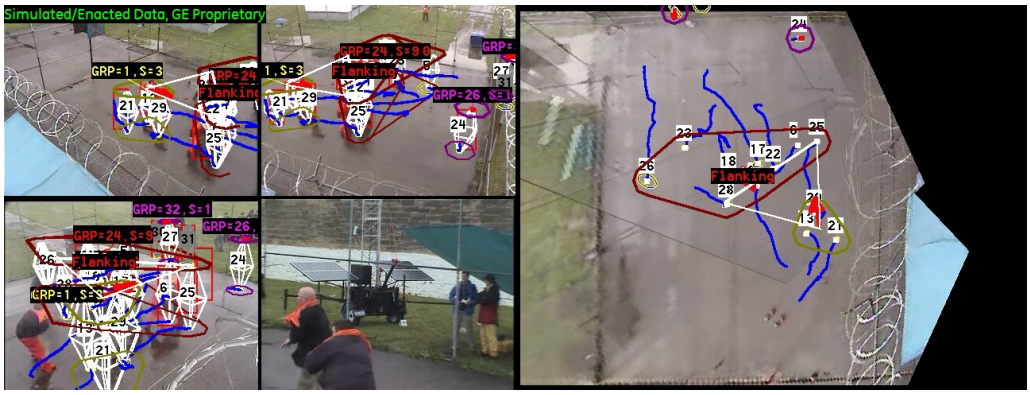


Figure 57: Flanking Detection. Two examples where the system correctly detected a flanking event. In both cases the flanking detector raised an alert shortly before fights between groups of inmates broke out. In (a) one can see the (fake) improvised weapons (bricks) that the attacking inmates are carrying. The group surrounded by a red polygon (ID=36) is flanking a second group visualized in yellow (ID=61). In (b) a large group (ID=24) is flanking a second one (ID=1).

8.7 Performance Evaluation

To summarize the overall performance of the system at the Mock Prison Riot, we systematically processed all sequences in the table listed in Appendix C.2 and summarized the results below. The following sections list the title of the sequence, a brief description of the scenario/event, whether the event was successfully detected by the system and a brief description of how the system performed. It should be noted that this analysis was based on a qualitative analysis of all low-level events detected by the system for every sequence. The actual fusion of all low level events into semantically higher level scenario detectors is the goal of a current NIJ research effort. It should also be noted that the following analysis was performed offline on the data collected at the MPR.

8.7.1 Sequence "Utah Leader Attack" (Nr. 00)

Event Description: *Two groups of prisoners in recreational yard. One group is covertly approaching the other. Member of circeling group attacks (stabs) leader of second group.*
Detected: Yes
Processing Result: The system detects the presence of two distinct groups and a "flanking" event. The approach of the encirceling group is detected as well as the short fight between the attacker and her victim. The attacker is detected as a fast moving person. The system detects the event as well predict the onset of the aggression.

8.7.2 Sequence "Utah Leader Attack 2" (Nr. 01)

Event Description: *Two groups of prisoners in recreational yard. One group B is playing basketball in the vicinity of the other group A. Suddenly one member from B quickly attacks a single individual (the leader) in A. A quick dispersion follows. There is very little agitation in the video.*
Detected: Yes
Processing Result: The system detects the fact that two distinct groups are present and the actual aggression in the video. The system is not able to make a reliable prediction of the event before it occurs.

8.7.3 Sequence "Gang Killing other Gang" (Nr. 02)

Event Description: *Two groups of prisoners in recreational yard. One group A is playing basketball. A second group B slowly approaches. Group B attacks the entire group A. Group A is down and group B quickly disperses.*
Detected: Yes
Processing Result: The system detects the aggression at the time it occurs. The attacks happens

very suddenly and quickly and the system is not able to make a reliable prediction of the event *before* it occurs.

8.7.4 Sequence "Gang Killing other Gang 2" (Nr. 03)

Event Description: *Two groups of prisoners in recreational yard. One group A is playing basketball. A second group B slowly approaches, picking up contraband and makeshift weapons. Group B attacks the entire group A. Group A is down and group B quickly disperses.*
Detected: No
Processing Result: The basketball playing of group A is causing multiple "aggression" false alarms. The system does detect the approach of the second group. A detector that would detect the act of basketball playing in order to separate basketball from standard aggression/agitation would solve this false alarm problem.

8.7.5 Sequence "Gang Killing other Gang 3 - Unrehearsed" (Nr. 04)

Event Description: *One group A is discussing something in the recreational yard. A second group B slowly approaches. Group B attacks the entire group A. Part of group A is down. One member of A struggles longer and is being chased by B and is also downed. Group B quickly disperses.*
Detected: Yes
Processing Result: The system clearly detects the presence of two distinct groups, the approach of B and the final attack. The system both predicts and detects the occurrence of violence.

8.7.6 Sequence "Aborted Attack" (Nr. 05)

Event Description: *Two groups are playing in the recreational yard. One group A is playing basketball while a second group B is approaching. Group A notices the approach and goes into a defensive position. The groups face each other but no fight occurs. Group B is aborting the planned attack.*
Detected: N/A
Processing Result: The system clearly detects the presence of two separate groups and detects an unusual amount of "approaching" motion patterns. No event that indicates a possible event is triggered.

8.7.7 Sequence "Aborted Attack 2" (Nr. 06)

Event Description: *Two groups are playing in the recreational yard. One group A is playing basketball while a second group B is approaching. Group A notices the approach and goes into a defensive position. The groups face each other and some pushing and shoving occurs. Group B is aborting the planned attack.*
Detected: N/A
Processing Result: The system clearly detects the presence of two separate groups *and* detects an unusual amount of "approaching" motion patterns. Several flanking events are triggered which are predictors of a possible aggression event.

8.7.8 Sequence "Gang Argument - Prisoners get attacked" (Nr. 07)

Event Description: *Two groups A and B are hanging out in the recreational yard. Two members from A and B are discussing something. They separate. Group A is unhappy and plans an attack. They pick up rocks and weapons and approach group B who is backed into a corner. A fight ensues.*
Detected: Yes
Processing Result: The system detects the approach and a flanking event. It hence is able to *predict* the onset of the fight. It furthermore detects the fight.

8.7.9 Sequence "Gang Initiation" (Nr. 08)

Event Description: *Two gangs A and B in rec yard. An unaffiliated person c (to be initiated into group B) talks to person a from group A. Group B attacks person a. Person c tries to block group B from helping. Fight breaks out between group A and B. Lots of agitation.*
Detected: Yes
Processing Result: The system detects the presence of separate gangs. After detecting the fact that c is moving fast, the system detects the agitation. A multitude of fight events are triggered.

8.7.10 Sequence "Contraband Exchange" (Nr. 09)

Event Description: *Two people a and b from groups A and B meet and exchange contraband.*
Detected: Part.
Processing Result: The system detects the approach between the inmates, but this is not an unusual event to be detected. The system is in its current form not able to flag this event as an unusual one.

8.7.11 Sequence "Multiple Contraband Exchange" (Nr. 10)

Event Description: *Two groups A and B meet (walk through each other), while exchanging items.*
Detected: Part.
Processing Result: The system detects a series of unusual approaches and many structural changes in terms of the grouping of the people in the scene. This should be exploited in the future to detect such an event. In its current form the system does not flag this as an unusual event.

8.7.12 Sequence "Contraband with Fight" (Nr. 11)

Event Description: *Two members from two groups exchange contraband. They are unhappy about the exchange. A fight ensues.*
Detected: Yes
Processing Result: After detecting distinct groups, the system also detects the fights.

8.7.13 Sequence "Blended Transaction" (Nr. 12)

Event Description: *Contraband is exchanged multiple times and finally a person in the rec yard stabbed. The event happens very quickly.*

Detected: No
Processing Result: The stabbing is so covert that the system does not detect it.

8.7.14 Sequence "Shanking followed by Leaving" (Nr. 13)

Event Description: *A group is playing basketball. They invite a gang-outsider to join in. He's happy to accept the invitation and is about to throw a ball but is then stabbed like Julius Caesar in the roman senate.*
Detected: Yes
Processing Result: Aside from several agitation false alarms due to aggression, the system detects the actual stabbing event. There is also a distinct crowding around the victim during the stabbing that currently is not modeled as an unusual event.

8.7.15 Sequence "Gang Hanging Out Followed By Several Fights" (Nr. 14)

Event Description: *Gangs are hanging out in the rec yard. Several fights suddenly break out.*
Detected: Yes
Processing Result: Aside from a small false alarm caused by the shaking fence, the event is detected successfully.

8.7.16 Sequence "Fight Followed by Guards Leading Offender Off" (Nr. 15)

Event Description: *A basketball game turns into a fight. A guard intervenes and leads the offender off.*
Detected: Yes
Processing Result: Two aggression events are successfully detected: the initial attack and the guard struggling with the inmate.

8.7.17 Sequence "Fight Followed by Guards Leading Offender Off" (Nr. 16)

Event Description: *Contraband is exchanged followed by attack. Guard intervenes and apprehends inmate. There is a struggle between the guard and the inmate.*
Detected: Yes
Processing Result: Both aggression events are detected successfully.

8.7.18 Sequence "Contraband – Officer Notices" (Nr. 17)

Event Description: *Officer notices an exchange of contraband. He secures the yard and but one prisoner makes a wrong move and is wrestled down by the guard.*
Detected: Yes
Processing Result: The initial contraband handoff is detected as "agitation". So is the act of securing the yard (small struggles) and the larger struggle with the inmate at the ends.

8.7.19 Sequence "Argument Between Gangs – Officer Assault" (Nr. 18)

Event Description: *A basketball game turns violent. A guard intervenes and gets attacked.*
Detected: Yes
Processing Result: System detects agitation a few seconds before the fight (i.e., predicting the event) and the attack on the officer.

8.7.20 Sequence "Contraband exchange followed by guard searching inmates" (Nr. 19)

Event Description: *Contraband is exchanged, which a guard notices (hard to see from video). He secures the yard and searches everyone. There is no aggression.*
Detected: N/A
Processing Result: The system is not designed to detect contraband exchange. Beside the exchange, nothing unusual happens and not event is detected.

8.7.21 Sequence "Prisoner being attacked and guard intervening" (Nr. 20)

Event Description: *A prisoner is stabbed by fellow inmates during a yard exercise.*
Detected: No
Processing Result: The event is very subtle and is missed by the system.

8.7.22 Sequence "Fight breaking out between gang members and officers breaking it up" (Nr. 21)

Event Description: *A fight between gang members.*
Detected: Yes
Processing Result: The system detects the aggression during the fight. It does not predict the onset of the event.

8.7.23 Sequence "Fight between gangs. Guards breaking fight up" (Nr. 22)

Event Description: *Two gangs face off for a little while before a fight ensues.*
Detected: Yes
Processing Result: The system detects the distinct groups as well as the approach and final aggression. It hence predicts the event successfully.

8.7.24 Sequence "Fight between gangs. Guards breaking fight up" (Nr. 23)

Event Description: *Two gangs face off for a little while before a fight ensues.*
Detected: Yes
Processing Result: The system detects the distinct groups as well as the aggression.

8.7.25 Sequence "Gangs fighting. Guards breaking fight up." (Nr. 24)

Event Description: *Two gangs approach and fight immediately.*
Detected: Yes

Processing Result: The system detects the onset of the aggression but there is not sufficient time for the system to make a prediction.

8.8 Summary

The results of the above performance analysis is tabulated below.

Nr.	Title	Detected?	Predicted?
00	Utah Leader Attack	Yes	Yes
01	Utah Leader Attack 2	Yes	No
02	Gang Killing other Gang	Yes	No
03	Gang Killing other Gang 2	No	N/A
04	Gang Killing other Gang 3 - Unrehearsed	Yes	Yes
05	Aborted Attack	N/A	N/A
06	Aborted Attack 2	N/A	N/A
07	Gang Argument - Prisoners get attacked	Yes	Yes
08	Gang Initiation	Yes	No
09	Contraband Exchange	Part.	No
10	Multiple Contraband Exchange	Part.	No
11	Contraband with Fight	Yes	No
12	Blended Transaction	No	No
13	Shanking followed by Leaving	Yes	No
14	Gang Hanging Out Followed By Several Fights	Yes	No
15	Fight Followed by Guards Leading Offender Off	Yes	N/A
16	Fight Followed by Guards Leading Offender Off	Yes	N/A
17	Contraband – Officer Notices	Yes	N/A
18	Argument Between Gangs – Officer Assault	Yes	Yes
19	Contraband exchange followed by guard searching inmates	N/A	N/A
20	Prisoner being attacked and guard intervening	No	No
21	Fight breaking out between gang members and officers breaking it up	Yes	No
22	Fight between gangs. Guards breaking fight up	Yes	Yes
23	Fight between gangs. Guards breaking fight up	Yes	No
24	Gangs fighting. Guards breaking fight up.	Yes	N/A

One can see that out of 25 sequences collected the system successfully detected 17 events and encountered 3 failures and 2 partial detections.

Nr of sequences:	25
Successful detections:	17
Missed detections:	3
Partial detections:	2
Successful predicitons:	5
Not applicable:	3

9 Conclusions

The data collection at the MPR and subsequent analysis has shown that the system has an about 70% chance of detecting the occurrence of disorderly or aggressive events in the observed prison environment and currently a 20% chance of predicting the event *before* it occurs.

It should be noted that the analysis of the MPR data was performed *offline*, after several of the event detection system was optimized and adapted to the MPR data.

9.1 Law Enforcement Feedback

After every data collection and testing session we solicited feedback from law enforcement officers with respect to the merit and performance of our proposed technology. The general feedback was very positive and the correctional officers where enthusiastic about what the technology is able to do for them.

Figure 58: Feedback. The GE Global Research team actively solicited comments and feedback after scenarios. The response was overall very positive. Independent evaluation is also being conducted by the WVHTF in their "Mock Prison Riot Operational Assessment Report".

The following key points where made by the officers:

- The technology is very relevant to correctional operations.
- Automated surveillance in general very desired because manually monitoring all areas of a prison is difficult and expensive.
- The ability to detect the onsets of fights is a highly desired capability.
- Detection of contraband exchange is of high priority.

- Social network analysis is highly desirable for some prison settings. Some larger prisons have gang intelligence officers devoted to tracking gang associations over time. Such venues would like to monitor subjects in the eating area where a lot of interaction between and within gangs occur.
- Some activity is very elusive and requires a high level of comprehension on behalf of the observer (and hence also on behalf of an analytics system).
- The automatic PTZ control got high marks from the officers. To be able to recognize in detail which prisoners are part of certain activities is of high importance.
- Gang membership is often expressed via Tattoos and Clothing. Some officers wondered if the video system would be able to detect this (currently it is not).
- Cost and installation complexity are a concern. Some prisons have little video infrastructure.

9.2 Observations

In addition to direct feedback, a number of insights where gained during the Mock Prison Riot data collection and subsequent testing.

- The system setup for the event (operator console outdoor under tarp and location in the back of recreational yard) made it difficult to operate the system (in rain and bright light) and demonstrate the technology to the practitioner community. In the future, a better location and approach for shelter should the sought (e.g., trailer, camper or large tent).
- The video analytics system needs to become more convenient in order to let law enforcement practitioners try the system on site and help future adoption.
- Event notification has to become more concise. The current system has a tendency to overload the operator with too many visualizations and event notifications.
- The PTZ camera should be slaved to the event detection system. Whenever an event is detected, the PTZ camera should zoom in to obtain a shot of the action. At the moment the PTZ is systematically trying to capture faces, which is not an ideal mode of operation when threatening events occur.
- The *official* MPR scenarios as organized in 2009 are not a good match for testing the developed technology. The focus of the MPR is the *response* to riotous events. The responses often involve tens of officers overwhelming inmates within only a few seconds of activity. This crowd density and the use of tear gas and flash-bang simulants make it difficult for the system to detect useful events, other than the presence of large crowds and strong agitation. In the future we will request dedicated MPR scenarios that feature more of the prisoner behavior leading up to the aggression/fight/misconduct.

- The current system is not able to *directly* detect contraband exchanges. Future studies should investigate this problem in more detail because it is of high priority to corrections officers.
- Certain causes for false alarms should be tacked. The most important once being the swaying of the fence (the movement of barb wire where interpreted by the system as aggressive/agitated motion in the scene) and basketball play.
- More normalcy data is needed. Most videos captured actual events happening. To make accurate assessments about detection rates, a large amount of *benign* footage needs to be available.

9.3 Law Enforcement Relevance and Impact

We have completed the development and implementation of a number of computer vision algorithms that have the potential to greatly help law enforcement officers monitor places such as prisons, schools, parks and public venues.

- The crowd parameter estimation presented in Section 5.3 has the ability to monitor public places and to alert authorities of large gatherings or when crowd density has reached a dangerous level.
- The motion pattern detection algorithms described in Section 5.4 can automatically detect FOLLOWING and CHASING events, which can help to keep public parks safe and automatically detect suspicious events in correctional facilities.
- The aggression detection algorithms described in Section 5.7 and 5.8 can automatically detect the occurrence of agitation and aggression. Aggression detection can be used to quickly dispatch officers to the scene of the event and also, if available, aim additional PTZ camera assets to obtain higher resolution footage of the event.
- The PTZ targeting system presented in Section 6.1 is a spin-off capability developed under this program. The algorithm was developed to enable the capture of faces for the social network analysis, but is useful far beyond that particular application. Automatic PTZ targeting can be leveraged to capture faces (see Section 6.2) and high-resolution footage from a

All Mock Prison Riot data was *enacted* by law enforcement and corrections practitioners.

distance of subjects involved in any kind of suspicious activities without requiring operators to manually aim the camera. The developed system intrinsically strives for optimal capture strategies, which directly translates to potential savings in labor, time and equipment when compared to alternative ways to capture such imagery.

- Our research into detecting and recognizing faces from PTZ cameras has important applications for law enforcement as it describes methods for performing face recognition of uncooperative individuals from a distance.

- Our work is the first one to show that one can recover social network structure automatically from video. This can have an important impact on law enforcement. The social network analysis algorithm presented in Section 7 allows officers to automatically estimate social network structures and determine things such as group-size, membership and leadership structure automatically from video. The system has been shown to provide promising results and could be used in environments such as prison dining-halls, recreational yards and other environments, where law enforcement and corrections would like to automatically gather gang-intelligence and where such tasks nowadays have to be performed manually through careful observation.

Overall we have shown that the system developed under our NIJ program has the ability to detect law enforcement and corrections relevant behaviors during the 2009 Mock Prison Riot. It was demonstrated that behaviors that are predictive of fights can be found and utilized in prison environments. This will allow correctional officers to prevent the event before it occurs, which will keep officers and inmates safer and cost down.

9.4 Next Steps

In the follow up program "Advanced Behavior Recognition in Crowded Environments" (2009-SQ-B9-K013) we will focus on improving the behavior recognition of the system based on the low-level motion pattern event detectors developed under this program. We also plan to gauge the

performance of our system in real-world field testing through deployment at local law enforcement facilities. Currently the system is tested for crowds with moderate densities. For high density crowd, we will adopt a divide-and-conquer approach to deal with sub-groups of smaller sizes separately.

All Mock Prison Riot data was *enacted* by law enforcement and corrections practitioners.

Appendices

A Public Dissemination

As part of this research program we have disseminated our work through the following papers:

[1] Nils Krahnstoever, Ting Yu, Ser-Nam Lim, Kedar Patwardhan, Peter Tu, "Collaborative Real-Time Control of Active Cameras in Large-Scale Surveillance Systems", Workshop on Multi-camera and Multi-modal Sensor Fusion Algorithms and Applications (M2SFA2), Marseille, France, October 18, 2008.

[2] Ting Yu, Ser-Nam Lim, Kedar Patwardhan, Nils Krahnstoever, "Monitoring, Recognizing and Discovering Social Networks", IEEE Conference on Computer Vision and Pattern Recognition, 2009.

[3] Nils Krahnstoever, Ting Yu, Ser-Nam Lim, Kedar Patwardhan, "Collaborative Real-Time Control of Active Cameras in Large-Scale Surveillance Systems", in "Multi-Camera Networks: Concepts and Applications", Editors: Hamid Aghajan and Andrea Cavallaro, Elsevier, 2009.

The following provisional patent was filed:

Nils Krahnstoever, Ting Yu, Ser-Nam Lim, Kedar Patwardhan, Peter Tu, "Collaborative real-time control of active cameras in large scale surveillance systems", Provisional Filing, US 61/091581.

B Reviews and Meetings

B.1 Technical Working Group Meeting

Nils Krahnstoever (GRC) presented an overview of the porgram at the Technical Working Group Workshop in Orlando, Florida on October 25, 2007. The presentation was well received and we established valuable contacts to practitioners from various areas of law enforcement, to William Ford as well as former program manager Patty Wolfhope.

B.2 Kick-Off Meeting at NIJ

Nils Krahnstoever (PI), Ting Yu and Peter Tu presented an overview of the current program to Francis Scott, Sensors and Surveillance Portfolio Manager, at the National Institute of Justice on May 16th 2008. The meeting served both as a kick-off meeting for the current program as well as an opportunity to present some of the GE Global Research background work and future plans in the area security and surveillance.

B.3 Sensor and Surveillance Center of Excellence Visit

On June 6th, 2008, Chris McAleavey (Deputy Director National Sensors, Surveillance and Biometric Technologies Center Of Excellence), visited the GE Global Research Center. He was given a detailed update on this and various other NIJ programs run by GE Global Research. Chris in particular visited the three GE Global Research Testbeds (at the research center, Albany Airport and a local retail store). The entire meeting was very positive and especially the retail testbed received praise for being very relevant for the type of law enforcement challenges and applications addressed by this program.

B.4 2008 Technologies for Critical Incident Preparedness Expo (TCIP)

Nils Krahnstoever (PI) visited the TCIP Expo in Chicago, IL and presented current results and findings to the law enforcement community. The exhibition in the form of a booth was very successful and visited by a large number of law enforcement as well as non-law enforcement customers. The presentation and exhibit material was received favourably, and several useful connections with the community have been established.

B.5 Mock Prison Riot 2009

Nils Krahnstoever (PI) and Ting Yu performed a preliminary visit of the Mock Prison Riot site in Moundsville, WV in March 2009.

All Mock Prison Riot data was *enacted* by law enforcement and corrections practitioners.

The main Mock Prison Riot Event took place in May. Nils Krahnstoever and Ting Yu where accompanied by Don Hamilton, a member of the Visualization and Computer Vision lab, who has extensive experience in performing site installations.

B.6 IEEE Conference on Computer Vision 2009

Nils Krahnstoever attended the IEEE Conference on Computer Vision and Pattern Recognition in Miami, FL to present the paper “Monitoring, Recognizing and Discovering Social Networks”.

C Mock Prison Riot Data

C.1 Data Recorded while Processing

Nr	id	name	description	tags	frames	duration	time
1	1232	013. MPR Sunday First Test	Utah Team: Getting ready (signing and dressing).	sequence-of-interest	55327	0h 52m 57s 798ms	2009-05-03 10:49:36.671
2	1237	014. MPR Sunday - Utah Team - People Walking Around	Utah Team: Walking around.	nij-enactment, walking-around	1636	0h 4m 54s 254ms	2009-05-03 11:45:42.312
3	1242	015. MPR Sunday - Utah Team - People Walking Around	Utah Team: Hanging out. Two gangs taking turn playing basket ball.	nij-enactment, gangs, hanging-out	6902	0h 10m 1s 722ms	2009-05-03 11:52:53.703
4	1247	016. MPR Sunday - Utah Team - Two Group Arguing	Utah Team: Two gangs arguing. Circeling. Pushing and Showing. Team leaders start getting aggravated. Gangs pull leaders apart.	nij-enactment, gangs, arguing, pushing	6156	0h 9m 20s 211ms	2009-05-03 12:03:38.875
5	1252	017. MPR Sunday - Utah Team - Leader Being Attacked	Utah Team: Two gangs. Leader being attacked (stabbed in back).	nij-enactment, gangs, attack, stabbing	4165	0h 6m 4s 730ms	2009-05-03 12:16:27.250
6	1257	017. MPR Sunday - Utah Team - Leader Being Attacked	Utah Team: Inmate is attacked (stabbed). Others move away after hit. Stabbing is very quick.	nij-enactment, gangs, attack, stabbing, move-away	2015	0h 2m 3s 34ms	2009-05-03 12:23:05.375
7	1262	018. MPR Sunday - Utah Team - Gang Killing Other Gang	Utah Team: One gang is attacking another (three people get stabbed). Attacking gang is slowly approaching other gang (whose members are playing basketball).	nij-enactment, gangs, attack, gang-on-gang, slow-approach	3439	0h 3m 59s 360ms	2009-05-03 12:31:23.687
8	1267	018. MPR Sunday - Utah Team - Gang Killing Other Gang	Utah Team: One gang plays basketball. Other gang retrieves hidden weapons. Attacks first gang. After the scenario, Nils is meeting with one gang in court yard, while the other gang is *unexpectedly* sneaking up and attacks again. The first gang did not know about this.	nij-enactment, gangs, attack, gang-on-gang, hidden-weapons, unexpected	4461	0h 5m 23s 19ms	2009-05-03 12:36:34.109
9	1272	019. MPR Sunday - Utah Team - Aborted Attack	Utah Team: One gang is almost attacking another. Gang is noticing approach and goes into defensive position. No attack occurs.	nij-enactment, gangs, almost-attack, slow-approach, defensive	2299	0h 2m 54s 856ms	2009-05-03 12:49:10.062
10	1277	019. MPR Sunday - Utah Team - Aborted Attack	Utah Team: One gang is almost attacking another. Gang is noticing approach while playing basketball and goes into defensive position. Pushing and shoving. No serious fighting.Camera 0: First gang entering view just after frame 400. Good view to evaluate agitation detection (session 1278, recorded session 85 Ű frame 79600).	nij-enactmentgangs, almost-attack, pushing, slow-approach, defensive	3207	0h 4m 13s 210ms	2009-05-03 12:52:56.046
11	1282	020. MPR Sunday - Utah Team - Face Capture	Utah Team: Capture of all faces. Team is walking toward PTZ camera.	nij-enactment, face-capture	12933	0h 11m 15s 570ms	2009-05-03 12:59:26.125
12	1732	029. MPR Monday - Lake Erie MPR - 11AM	Erie scenario Monday 11am.	crowded, smoke, scenario	8429	0h 9m 55s 849ms	2009-05-04 10:57:43.328

All Mock Prison Riot data was *enacted* by law enforcement and corrections practitioners.

13	1752	030. MPR Monday - Lake Erie - Gangs Hanging Out	Gangs hanging out. Argument. Followed by prisoner being attacked.	nij-enactment, gangs, attack	3845	0h 3m 55s 257ms	2009-05-04 11:49:48.171
14	1757	031. MPR Monday - Lake Erie - Gangs Initiation	Two gangs. One person is chosen to attack member of other gang.	nij-enactment, gangs, attack	1927	0h 1m 59s 329ms	2009-05-04 11:55:54.187
15	1762	032. MPR Monday - Lake Erie - Contraband Exchange	Exchange of contraband.	nij-enactment, gangs, contraband	1726	0h 1m 52s 322ms	2009-05-04 12:00:12.687
16	1767	033. MPR Monday - Lake Erie - Multiple Contraband Exchange	Contraband exchange between mutliple people in two gangs.	nij-enactment, gangs, multiple-contraband	1348	0h 1m 21s 789ms	2009-05-04 12:02:33.250
17	1772	034. MPR Monday - Lake Erie - Contraband with Fight	Contraband exchange followed by argument over item with finally a fight breaking out.	nij-enactment, gangs, contraband, fight, gang-on-gang	1614	0h 1m 38s 406ms	2009-05-04 12:04:45.765
18	1777	035. MPR Monday - Lake Erie - Blended Transaction	Transaction during basketball game followed by attack.	nij-enactment, gangs, contraband, attack	1850	0h 2m 4s 135ms	2009-05-04 12:08:39.656
19	1782	036. MPR Monday - Lake Erie - Shanking followed by Leaving	Person is invited to participate in basketball game, but then stabbed by group.	nij-enactment, gangs, attack, sneaky	2370	0h 2m 30s 830ms	2009-05-04 12:12:18.359
20	1787	037. MPR Monday - Lake Erie - Face Captures	Lake Erie team showing faces to camera.	nij-enactment, face-capture	5401	0h 4m 35s 199ms	2009-05-04 12:22:28.343
21	1797	038. MPR Monday - North Dakota State Penitentiary - 1PM Scenario	Prisoner stabbed inmate and holding him hostage. Negotation fails. SWAT team secures situation.	scenario	7267	0h 8m 41s 936ms	2009-05-04 12:54:15.781
22	1847	040. MPR Monday - Montgomery County - 4PM Scenario	Montgomery County securing prisoners that refuse to go back to their cells. Line formation.	scenario, lawn	10733	0h 15m 14s 397ms	2009-05-04 16:02:54.468
23	1852	041. MPR Monday - Montgomery County - Test Run	Getting ready to collect scenarios with Montgomery County.	sequence-of-interest	4418	0h 5m 2s 764ms	2009-05-04 16:47:33.375
24	1857	042. MPR Monday - Montgomery County - Gang Hanging Out	Two gangs hanging out and then fighting followed by several fights breaking out.	nij-enactment, gangs, fight, multiple-fights	4071	0h 4m 29s 730ms	2009-05-04 16:53:01.171
25	1867	043. MPR Monday - Montgomery County - Assault on Prisoner	Fight breaking out between prisoners. Guard moving in and leading offender off.	nij-enactment, gangs, fight, guard	2792	0h 2m 25s 24ms	2009-05-04 17:02:47.890
26	1872	044. MPR Monday - Montgomery County - Assault with Weapons Passing	Contraband handoff followed by assault on prisoner.	nij-enactment, gangs, contraband, attack	3495	0h 3m 7s 104ms	2009-05-04 17:06:38.250
27	1877	045. MPR Monday - Montgomery County - Drug Passing	Contraband, officer noticing and starting to search. Contraband walks out of yard.	nij-enactment, contraband, gangs, fight, guard	1807	0h 1m 53s 957ms	2009-05-04 17:11:39.453
28	1927	047. MPR Tuesday - Hazelton - Scenario 10am	Hazelton scenario.	scenario	7940	0h 12m 3s 355ms	2009-05-05 09:50:08.421
29	1937	048. MPR Tuesday - North Lawn - Scenario 1045AM	Bomb busting scenario (long and not very sccessful).	lawn, scenario	24427	0h 40m 55s 811ms	2009-05-05 10:49:22.156
30	1982	052. MPR Tuesday - North Yard - Test Run	Preparing with volunteers.	nij-enactment	1488	0h 1m 11s 712ms	2009-05-05 13:44:24.171
31	1987	053. MPR Tuesday - Delaware County - Officier Gets Assault	Argument between gangs. Guard intervening. Guard being attacked.	nij-enactment, gangs, argument, guard, attack-on-guard	2457	0h 2m 4s 503ms	2009-05-05 13:46:53.468
32	1992	054. MPR Tuesday - Delaware County - Countraband Exchange	Contraband exchange, followed by guard searching inmate.	nij-enactment, gangs, contraband, guard	1882	0h 1m 40s 911ms	2009-05-05 13:50:33.406
33	1997	055. MPR Tuesday - Delaware County - Shanking	Prisoner being attacked. Guard intervening.	nij-enactment, gangs, attack, guard, intervention	2309	0h 2m 30s 597ms	2009-05-05 13:52:50.343

34	2042	057. MPR Tuesday - Chatham County - Scenario 230pm	Chatham County scenario. Prisoners acting up. Moving in to secure prisoners.	scenario	4520	0h 5m 16s 879ms	2009-05-05 14:26:28.281
35	2047	058. MPR Tuesday - Chatham County - Redman - Scenario 230pm	Chatham County redman suite. Arresting prisoners in force.	scenario	10389	0h 13m 35s 590ms	2009-05-05 14:39:52.109
36	2052	059. MPR Tuesday - Chatham County - Redman - Scenario 230pm	Chatham County single prisoner being secured (red man suite).	scenario	1196	0h 1m 35s 337ms	2009-05-05 14:54:38.671
37	2062	061. MPR Tuesday - Naval - Scenario 330pm	Naval scenario. Ten prisoners. Apprehending prisoners.	scenario, smoke	5933	0h 8m 58s 187ms	2009-05-05 15:27:02.375
38	2067	062. MPR Tuesday - Chatham County - Fight	Fight breaking out between gang members. Officers breaking it up.	nij-enactment, gangs, fight, guard, intervention, gang-on-gang	4266	0h 3m 36s 802ms	2009-05-05 15:48:12.625
39	2072	063. MPR Tuesday - Chatham County - Gang Fight	Fight between gangs. Guards breaking fight up.	nij-enactment, gangs, fight, guard, intervention, gang-on-gang	2019	0h 1m 51s 855ms	2009-05-05 15:53:11.250
40	2077	064. MPR Tuesday - Chatham County - Stabbing	Two gangs. Leaders meet. One leader gets attacked from behind. Guards securing situation.	nij-enactment, gangs, fight, guard, intervention, gang-on-inmate	2666	0h 2m 13s 579ms	2009-05-05 15:55:46.312
41	2082	065. MPR Tuesday - Chatham County - Two Gangs Fighting	Gangs fighting. Guards breaking fight up.	nij-enactment, gangs, fight, guard, intervention, gang-on-gang	1809	0h 1m 38s 507ms	2009-05-05 15:58:34.312
42	2117	068. MPR Wednesday - Federal Bureau of Prison - 10am Scenario	Inmates playing dodge ball. Group meeting. Inmates refusing to leave. Guard team moves in an apprehends prisoners. Scenario repeated multiple times.	scenario, multiple	15642	0h 24m 15s 788ms	2009-05-06 09:51:50.468
43	2161	069. MPR Wednesday - Lubbock County - 11am Scenario	Inmates playing dodge ball. Group meeting. Inmates refusing to leave. Guard team moves in an apprehends prisoners.	scenario, smoke	7697	0h 18m 54s 503ms	2009-05-06 10:57:18.515
44	2166	069. MPR Wednesday - Lubbock County - 11am Scenario	Photo ops.	sequence-of-interest	13727	0h 16m 42s 961ms	2009-05-06 11:16:22.656

C.2 Sequences Processed in Detail

Nr	id	name	description	tags	frames	duration	time
0	1252	017. MPR Sunday - Utah Team - Leader Being Attacked	Utah Team: Two gangs. Leader being attacked (stabbed in back).	nij-enactment, gangs, attack, stabbing	4165	0h 6m 4s 730ms	2009-05-03 12:16:27.250
1	1257	017. MPR Sunday - Utah Team - Leader Being Attacked	Utah Team: Inmate is attacked (stabbed). Others move away after hit. Stabbing is very quick.	nij-enactment, gangs, attack, stabbing, move-away	2015	0h 2m 3s 34ms	2009-05-03 12:23:05.375
2	1262	018. MPR Sunday - Utah Team - Gang Killing Other Gang	Utah Team: One gang is attacking another (three people get stabbed). Attacking gang is slowly approaching other gang (whose members are playing basketball).	nij-enactment, gangs, attack, gang-on-gang, slow-approach	3439	0h 3m 59s 360ms	2009-05-03 12:31:23.687

3	1267	018. MPR Sunday - Utah Team - Gang Killing Other Gang	Utah Team: One gang plays basketball. Other gang retrieves hidden weapons. Attacks first gang. After the scenario, Nils is meeting with one gang in court yard, while the other gang is *unexpectedly* sneaking up and attacks again. The first gang did not know about this.	nij-enactment, gangs, attack, gang-on-gang, hidden-weapons, unexpected	4461	0h 5m 23s 19ms	2009-05-03 12:36:34.109
4	1272	019. MPR Sunday - Utah Team - Aborted Attack	Utah Team: One gang is almost attacking another. Gang is noticing approach and goes into defensive position. No attack occurs.	nij-enactment, gangs, almost-attack, slow-approach, defensive	2299	0h 2m 54s 856ms	2009-05-03 12:49:10.062
5	1277	019. MPR Sunday - Utah Team - Aborted Attack	Utah Team: One gang is almost attacking another. Gang is noticing approach while playing basketball and goes into defensive position. Pushing and shoving. No serious fighting.Camera 0: First gang entering view just after frame 400. Good view to evaluate agitation detection (session 1278, recorded session 85 Ű frame 79600).	nij-enactmentgangs, almost-attack, pushing, slow-approach, defensive	3207	0h 4m 13s 210ms	2009-05-03 12:52:56.046
6	1752	030. MPR Monday - Lake Erie - Gangs Hanging Out	Gangs hanging out. Argument. Followed by prisoner being attacked.	nij-enactment, gangs, attack	3845	0h 3m 55s 257ms	2009-05-04 11:49:48.171
7	1757	031. MPR Monday - Lake Erie - Gangs Initiation	Two gangs. One person is chosen to attack member of other gang.	nij-enactment, gangs, attack	1927	0h 1m 59s 329ms	2009-05-04 11:55:54.187
8	1762	032. MPR Monday - Lake Erie - Contraband Exchange	Exchange of contraband.	nij-enactment, gangs, contraband	1726	0h 1m 52s 322ms	2009-05-04 12:00:12.687
9	1767	033. MPR Monday - Lake Erie - Multiple Contraband Exchange	Contraband exchange between mutliple people in two gangs.	nij-enactment, gangs, multiple-contraband	1348	0h 1m 21s 789ms	2009-05-04 12:02:33.250
10	1772	034. MPR Monday - Lake Erie - Contraband with Fight	Contraband exchange followed by argument over item with finally a fight breaking out.	nij-enactment, gangs, contraband, fight, gang-on-gang	1614	0h 1m 38s 406ms	2009-05-04 12:04:45.765
11	1777	035. MPR Monday - Lake Erie - Blended Transaction	Transaction during basketball game followed by attack.	nij-enactment, gangs, contraband, attack	1850	0h 2m 4s 135ms	2009-05-04 12:08:39.656
12	1782	036. MPR Monday - Lake Erie - Shanking followed by Leaving	Person is invited to participate in basketball game, but then stabbed by group.	nij-enactment, gangs, attack, sneaky	2370	0h 2m 30s 830ms	2009-05-04 12:12:18.359
13	1787	037. MPR Monday - Lake Erie - Face Captures	Lake Erie team showing faces to camera.	nij-enactment, face-capture	5401	0h 4m 35s 199ms	2009-05-04 12:22:28.343
14	1857	042. MPR Monday - Montgomery County - Gang Hanging Out	Two gangs hanging out and then fighting followed by several fights breaking out.	nij-enactment, gangs, fight, multiple-fights	4071	0h 4m 29s 730ms	2009-05-04 16:53:01.171
15	1867	043. MPR Monday - Montgomery County - Assault on Prisoner	Fight breaking out between prisoners. Guard moving in and leading offender off.	nij-enactment, gangs, fight, guard	2792	0h 2m 25s 24ms	2009-05-04 17:02:47.890
16	1872	044. MPR Monday - Montgomery County - Assault with Weapons Passing	Contraband handoff followed by assault on prisoner.	nij-enactment, gangs, contraband, attack	3495	0h 3m 7s 104ms	2009-05-04 17:06:38.250
17	1877	045. MPR Monday - Montgomery County - Drug Passing	Contraband, officer noticing and starting to search. Contraband walks out of yard.	nij-enactment, contraband, gangs, fight, guard	1807	0h 1m 53s 957ms	2009-05-04 17:11:39.453

18	1987	053. MPR Tuesday - Delaware County - Officier Gets Assault	Argument between gangs. Guard intervening. Guard being attacked.	nij-enactment, gangs, argument, guard, attack-on-guard	2457	0h 2m 4s 503ms	2009-05-05 13:46:53.468
19	1992	054. MPR Tuesday - Delaware County - Countraband Exchange	Contraband exchange, followed by guard searching inmate.	nij-enactment, gangs, contraband, guard	1882	0h 1m 40s 911ms	2009-05-05 13:50:33.406
20	1997	055. MPR Tuesday - Delaware County - Shanking	Prisoner being attacked. Guard intervening.	nij-enactment, gangs, attack, guard, intervention	2309	0h 2m 30s 597ms	2009-05-05 13:52:50.343
21	2067	062. MPR Tuesday - Chatham County - Fight	Fight breaking out between gang members. Officers breaking it up.	nij-enactment, gangs, fight, guard, intervention, gang-on-gang	4266	0h 3m 36s 802ms	2009-05-05 15:48:12.625
22	2072	063. MPR Tuesday - Chatham County - Gang Fight	Fight between gangs. Guards breaking fight up.	nij-enactment, gangs, fight, guard, intervention, gang-on-gang	2019	0h 1m 51s 855ms	2009-05-05 15:53:11.250
23	2077	064. MPR Tuesday - Chatham County - Stabbing	Two gangs. Leaders meet. One leader gets attacked from behind. Guards securing situation.	nij-enactment, gangs, fight, guard, intervention, gang-on-inmate	2666	0h 2m 13s 579ms	2009-05-05 15:55:46.312
24	2082	065. MPR Tuesday - Chatham County - Two Gangs Fighting	Gangs fighting. Guards breaking fight up.	nij-enactment, gangs, fight, guard, intervention, gang-on-gang	1809	0h 1m 38s 507ms	2009-05-05 15:58:34.312

C.3 Data Recorded without Processing

Nr	id	name	description	tags	frames	duration	time
1	69	014. MPR Sunday First Test	Utah Team getting ready.	non-processed, master-session	24298	0h 19m 23s 170ms	2009-05-03 11:23:21.625
2	74	015. MPR Sunday - Second System Recording	Utah Team: two gangs hanging out.	non-processed, master-session, nij-enactment	4401	0h 8m 0s 987ms	2009-05-03 11:44:20.015
3	79	015. MPR Sunday - Second System Recording	Utah Team: two gangs hanging out.	non-processed, master-session, nij-enactment	3474	0h 6m 42s 596ms	2009-05-03 12:00:27.875
4	84	016. MPR Sunday - Second System Recording	Utah Team: various scenarios.	non-processed, master-session, nij-enactment	114360	1h 5m 19s 641ms	2009-05-03 12:07:55.671
5	129	020. MPR Monday - Second System Recording - Lake Erie MPR - 11AM	Lake Erie scenario.	non-processed, master-session, scenario, smoke, crowd, dense-crowd	17193	0h 9m 50s 510ms	2009-05-04 10:57:56.765
6	139	021. MPR Monday - Second System Recording - Lake Erie - Scenario Capturing	Lake Erie enactment. All scenarios.	non-processed, master-session, nij-enactment	99025	0h 56m 36s 103ms	2009-05-04 11:47:49.578
7	144	022. MPR Monday - Second System Recording - North Dakota State Penitentiary - 1pm	North Dakota hostage scenario.	non-processed, master-session, scenario	23517	0h 13m 25s 458ms	2009-05-04 12:50:26.781

All Mock Prison Riot data was *enacted* by law enforcement and corrections practitioners.

8	149	024. MPR Monday - Second System Recording - Montgomery County - 4PM Scenario	Montgomery County scenario.	non-processed, master-session, lawn, scenario	28447	0h 16m 14s 153ms	2009-05-04 16:02:04.687
9	154	025. MPR Monday - Second System Recording - Montgomery County - Scenario Capturing	Montgomery County enactments.	non-processed, master-session, nij-enactment	57293	0h 32m 40s 353ms	2009-05-04 16:45:54.500
10	169	027. MPR Tuesday - Second System Recording - Hazelton - 10AM Scenario	Hazelton scenario, prisoners refusing to leave yard. Police overwhelms them.	non-processed, master-session, scenario	21351	0h 12m 12s 375ms	2009-05-05 09:50:14.937
11	179	028. MPR Tuesday - Second System Recording - Team Preparation - 1040AM	Some police team planning scenario.	non-processed, master-session	7071	0h 4m 3s 913ms	2009-05-05 10:41:40.640
12	184	028. MPR Tuesday - Second System Recording - Team Preparation - 1040AM	Door busting scenario on North lawn.	non-processed, master-session, lawn	73653	0h 42m 0s 664ms	2009-05-05 10:48:29.859
13	199	031. MPR Tuesday - Second System Recording - 1:40 -	Delaware county - preparation for scenario collection.	non-processed, master-session	7516	0h 4m 16s 928ms	2009-05-05 13:40:37.046
14	204	032. MPR Tuesday - Second System Recording - Unknown Team - Scenario Capturing	Delaware county - various scenario enactments.	non-processed, master-session, nij-enactment	19465	0h 11m 7s 134ms	2009-05-05 13:45:29.890
15	214	034. MPR Tuesday - Second System Recording - Chatham County - 0230PM Scenario	Chatham County scenario.	non-processed, master-session, scenario	10134	0h 5m 48s 567ms	2009-05-05 14:25:44.156
16	219	035. MPR Tuesday - Second System Recording - Chatham County - Redman - 0230PM Scenario	Chatham County scenario with Redman suits.	non-processed, master-session, scenario	25028	0h 14m 18s 254ms	2009-05-05 14:39:18.890
17	224	035. MPR Tuesday - Second System Recording - Chatham County - Redman - 0230PM Scenario	Chatham County scenario with single man in Redman suits.	non-processed, master-session, scenario	3533	0h 2m 3s 810ms	2009-05-05 14:54:17.890
18	249	036. MPR Tuesday - Second System Recording - Naval - 0330PM Scenario	Naval Brigg scenario.	non-processed, master-session, scenario	16383	0h 9m 23s 148ms	2009-05-05 15:26:54.265
19	254	037. MPR Tuesday - Second System Recording - Chatham County - Scenario Capturing	Chatham County enacting various scenarios.	non-processed, master-session, nij-enactment	38962	0h 22m 14s 634ms	2009-05-05 15:38:29.843
20	274	040. MPR Wednesday - Second System Recording - Federal Bureau - 10AM Scenario	Federal Bureau scenario.	non-processed, master-session, scenario	42893	0h 24m 28s 521ms	2009-05-06 09:52:27.093
21	279	041. MPR Wednesday - Second System Recording - Lubbock County - 11AM Scenario	Lubbock County scenario.	non-processed, master-session, scenario	37989	0h 21m 43s 298ms	2009-05-06 10:57:21.312

D Techinical Details of the PTZ Camera Control

D.1 Problem Formulation

Given a surveillance site, let $\mathbf{C}_i^{\mathrm{f}}$ be the set of $\mathbf{N}^{\mathrm{f}}$ fixed cameras. The tracking system operates [10, 11] on these fixed cameras that are calibrated with respect to a site-wide metric coordinate system [14, 39]. We assume that a ground plane-based tracker observes *targets*[2] $\mathbf{O} = \{\mathbf{O}^i | i = 1, \dots\}$ during its operation. The state of a target at time t_j is given by $\mathbf{O}_j^i = (\mathbf{x}_i^j, \mathbf{v}_i^j)$ with ground plane location $\mathbf{x} = (x, y)$ and ground plane velocity $\mathbf{v} = (v_x, v_y)$. At time t_j we assume that the system observes targets with indices $\mathbf{S}^j = \{s_i^j | i = 1, \dots, \mathbf{N}^j\}$.

Furthermore, we assume we are given $\mathbf{N}^{\mathrm{p}}$ PTZ cameras, $\mathbf{C}_i^{\mathrm{p}}$, with projective geometries $\{\mathbf{P}_i^{\mathrm{p}} | i = 1, \dots, \mathbf{N}^{\mathrm{p}}\}$. The projective matrices are defined with respect to a *home* position. The Pan Tilt Zoom (PTZ) state of a camera is defined as $\mathbf{s} = (\phi, \theta, r)$, where ϕ and θ represent pan and tilt angles, and r denotes the zoom factor. The state $\mathbf{s} = (0, 0, 1)$ corresponds to the camera in its home position. Our system has the ability to instruct a PTZ camera to focus on an object of a certain size at a (x, y, z) world location, the details of which are omitted here due to size limitations.

We refer to the move and follow strategy for a PTZ camera as a *plan*. A plan for the PTZ camera $\mathbf{C}_i^{\mathrm{p}}$ is denoted as a set of tasks $\mathbf{E}_i = \left[e_i^1, e_i^2, \dots, e_i^{N_i^E}\right]$ where e_i^l is the index of the l^{th} target that will be visited by $\mathbf{C}_i^{\mathrm{p}}$. We assume that every target in the plan will be followed for a fixed amount of time Δt^{fo}. If two consecutive entries in the plan denote different targets, the PTZ has to spend a time Δt^{mv} to move from one target to another. This time is, in practice, dependent on the PTZ state of the camera when the camera starts moving and the PTZ state of the camera when it reaches the next target. This time can be predicted, but for ease of discussion, we assume it to be a constant. A plan for all PTZ cameras is denoted as $\mathcal{E} = [\mathbf{E}_1, \dots, \mathbf{E}_{\mathbf{N}^{\mathrm{p}}}]$. When a PTZ camera has reached a target, it will start to capture close-up imagery. We assume that every time a PTZ camera is following an individual, it obtains one capture. We denote the j^{th} capture of target $\mathbf{O}^i$ by $\mathbf{c}_{ij}$. The amount and qualities of captures significantly affect the success of a PTZ plan. In the

[2] usually humans, but a target could also be a vehicle or an abandoned object

following section we will define performance objective that determines the optimality of such a PTZ plan.

D.2 Objective Function

In our work, the purpose of PTZ camera captures is to provide high-quality close-up imageries of individuals observed by the system for later facial biometrics tasks. We assume that the success probability of these tasks is determined by the quality of captures. Hence, our performance objective, which aims to optimize the success probability of the tasks, is essentially defined based on a set of quality measures that are modeled by probabilistic distributions.

D.2.1 Quality Measures

In the following, we describe each of the quality measures.

View Angle: Since our interest is in facial biometrics, a camera looking at a person from behind or a top-down viewpoint brings little value. This viewpoint concern is characterized by the angle, α, between the normal of the face of a person and the line direction defined by the PTZ camera location and the face position of the person. To measure α, instead of using the normal of the face directly, which is difficult to estimate, we use the travel direction of the person. This substitution is in general valid since people normally look in the same direction which they are heading. The closer to 0 degree α is, the better the chance that the camera is able to capture the frontal face of the person. We model this quantity with a Gaussian defined as

$$q^{\text{vw}}(\alpha) = \exp^{-\frac{\alpha^2}{2\sigma_{vw}^2}}, \tag{6}$$

Figure 59 (right) shows the quality function of $q^{\text{vw}}(\alpha)$.

Target-Camera Distance: The quality of captured imageries generally degrades as the distance, d_{tc}, between the target and the camera increases. Hence, our second quality measure is

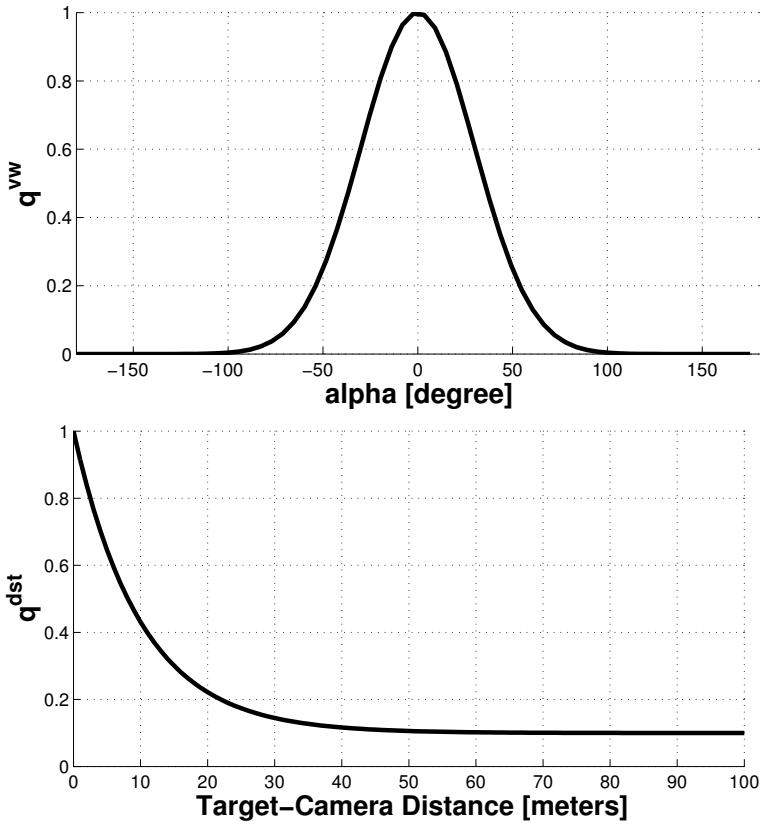


Figure 59: Quality Function of View Angle and Target to Camera Distance. Top: a Gaussian distribution is defined to model the capture quality as a function of view angle. Bottom: a two-component mixture model, which combines an exponential distribution and a uniform distribution, is defined to model the capture quality as a function of target to camera distance.

based on d_{tc} using a two-component mixture model

$$q^{\mathrm{dst}}(d_{tc}) = \gamma + (1-\gamma)\exp^{-\lambda d_{tc}}. \tag{7}$$

It combines an exponential distribution, which has a large mixture coefficient, with a uniform distribution as a small component. The exponential distribution models the trend that the probability of successfully capturing a target decreases as the target moves away from the camera, and the uniform distribution is to accommodate the slight chance of capturing a target even it is far away.

Target Trackability: The combined field of views of a fixed camera network defines a region that can be monitored by the system. Theoretically, any target inside this zone can be detected and tracked. Moreover, it is conceivable that a target outside the area may still have correct state estimates due to motion prediction. In reality, however, a target that is currently close to the zone boundary may have a higher probability of leaving the area soon, and therefore may have a smaller chance to be captured properly. By calculating the distance between the predicted location of the target and zone boundary, d_{tz}, where d_{tz} is positive if the target is inside the zone, and negative if outside, we obtain a probabilistic measure, $q^{\mathrm{trck}}(d_{tz})$, to quantify this target trackability. A mixture model of a logistic function and a uniform distribution is defined based on this distance as

$$q^{\mathrm{trck}}(d_{tz}) = \beta + (1-\beta)\frac{\exp^{\sigma_{trck}d_{tz}}}{1+\exp^{\sigma_{trck}d_{tz}}}. \tag{8}$$

Again the small uniform distribution serves as an outlier process to accommodate extreme conditions.

Target Reachability: Each PTZ camera has some mechanical limitation on its pan-tilt-zoom parameter range, defined by $(\phi_{min}, \phi_{max}, \theta_{min}, \theta_{max}, r_{min}, r_{max})$. If a target is on a location such that a PTZ camera has to set its parameter state outside this physical range, this capture would become impractical. Thus, a term, $q^{\mathrm{rch}}(\phi, \theta, r)$, which defines the target reachability from a PTZ camera is introduced. In our case, we simply take an uniform distribution for a PTZ parameter state that is within this range, and for all impractical states outside the range we set their reachability

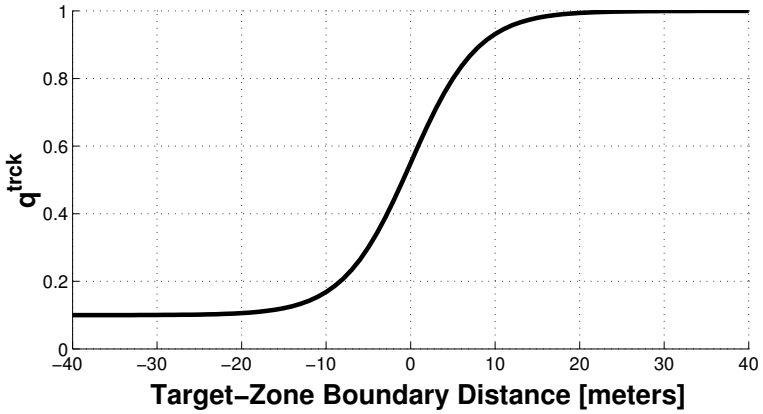


Figure 60: Quality Function of Target Trackability. A two-component mixture model, which combines a logistic function and a uniform distribution, is defined to model the capture quality as a function of target distance to the surveillance zone boundary.

probability to be 0:

$$q^{\mathrm{rch}}(\phi,\theta,r) \propto \begin{cases} 1, & (\phi,\theta,r) \in ([\phi_{min},\phi_{max}],[\theta_{min},\theta_{max}],[r_{min},r_{max}]) \\ 0, & \text{otherwise.} \end{cases} \tag{9}$$

D.2.2 Quality Objective

Assuming the statistical independence of individual quality measures, the overall quality q_{ij} of a capture $\mathbf{c}_{ij}$ is computed by the product of all four terms $q_{ij} = q^{\mathrm{vw}} q^{\mathrm{dst}} q^{\mathrm{trck}} q^{\mathrm{rch}}$.

As described above, the purpose of PTZ captures is to provide high quality input for certain biometrics tasks for individuals the system observes. Given a capture $\mathbf{c}_{ij}$ there is a probability p_{ij} that our task will succeed for target i (denoted as the event $S = \{\text{the biometrics task will succeed}\}$), which in turn is determined by the capture quality, and thus is modeled as a function of the quality q_{ij}. Hence, $p_{ij} = p(S|c_{ij}) = f(q_{ij})$. For the set of captures for target $\mathbf{O}^i$, $\mathbf{C}_i = \{\mathbf{c}_{ij} | j = 1,\ldots,\mathbf{N}_i^{\mathrm{c}}\}$, the probability of success (i.e., not failing every single time) is

$$p(S|\mathbf{C}_i) = \begin{cases} 0 & \text{if } \mathbf{N}_i^{\mathrm{c}} = 0 \\ 1 - \prod_j (1 - p(S|c_{ij})) & \text{otherwise} \end{cases} \tag{10}$$

and the total probability of success for capturing every target is

$$p(S) = \sum_i p(S|\mathbf{C}_i)p(\mathbf{C}_i) = \frac{1}{N}\sum_i \left[1 - \prod_j (1 - p(S|c_{ij}))\right]. \quad (11)$$

This overall success probability is an appealing metric. Its value increases as the number and quality of captures increase for every target if there is no PTZ resource constraint. However, in reality, due to the limited number of PTZ cameras and large amount of targets that may simultaneously exist in the surveillance area, the number of captures that each individual target can receive is often small. The overall quality of the captures measured over all targets in effect guides the competition for PTZ resources among the targets, where an optimal camera scheduling plan has to be found to maximally satisfy the biometrics tasks for all targets.

The optimization problem can now be defined as follows: Determine a plan $\mathcal{E}^*$ that optimizes the objective function (11) given the current site activity at time t_j, past captures and current PTZ camera activities (which may currently be executing a previously obtained plan).

D.2.3 Temporal Quality Decay

The predicted locations and view angles of the targets influence the quality measures and are dependent on the capturing time. Hence, the objective function (11) is determined by the order and times at which the PTZ cameras obtain the captures c_{ij}. As formulated above, targets that have had a long lifetime in the site will get captured multiple times, and the system spends less and less time on old targets in favor of new arrivals. This is a desired system behavior. In practice, however, this leads to problems because (i) human operators may feel that the system starts to neglect targets and (ii) if the tracker accidentally switches target ID, the system might actually neglect new or very recent arrivals. Hence, we add a temporal decay factor to discount the old captures of a target, which in turns allows the objective function to also take the re-capturing of the

target into consideration. We refine the capture quality evaluated at time t as follows

$$p_{ij}(t) = p(S|c_{ij}, t) = f(q_{ij})\tau(t - t_{ij}, \sigma_\tau, \Delta t_s, \Delta t_{cut}), \tag{12}$$

where t_{ij} is the time capture c_{ij} was taken and $\tau()$ is defined as

$$\tau(t - t_{ij}, \sigma_\tau, \Delta t_{cut}, \Delta t_s) = \begin{cases} 1 & \text{if } t - t_{ij} < \Delta t_s, \\ e^{-\frac{t - t_{ij} - \Delta t_s}{\sigma_\tau}} & \text{if } \Delta t_s \leq t - t_{ij} < \Delta t_{cut}, \\ 0 & \text{else.} \end{cases} \tag{13}$$

Within time Δt_s captures are rated fully, while after Δt_{cut}, they do not contribute to the objective function. In between, the quality factor decays with a rate determined by σ_τ.

D.3 Optimization

D.3.1 Asynchronous Optimization

Due to the time cost of determining PTZ plans, the plan optimization has to be performed asynchronously to the real-time tracking and PTZ control system. Our policy is that when we begin to optimize a new plan given the most recent states of the targets, say at time t_0, we predict the time, $t_0 + \Delta t_c$, when the computation will finish, and let all PTZ cameras finish up the tasks they will be executing at that time (see Figure 61 for illustration). This creates a predictable start state for the optimization and provides a simple approach for merging old and new PTZ plans.

During plan computation, the future locations of observed targets are predicted based on the expected elapsed time for moving the PTZ cameras and following the targets. The duration of a plan is given by the maximum completion time of all tasks in the plan. As the duration of the plan increases, state predictions become less reliable as targets start to deviate from their constant velocity path predictions. Due to this reason and in order to limit the computational complexity for computing a plan, only plans up to a certain completion time are explored. We call this maximum completion time the *planning horizon*, and define it to be the time when the first previously sched-

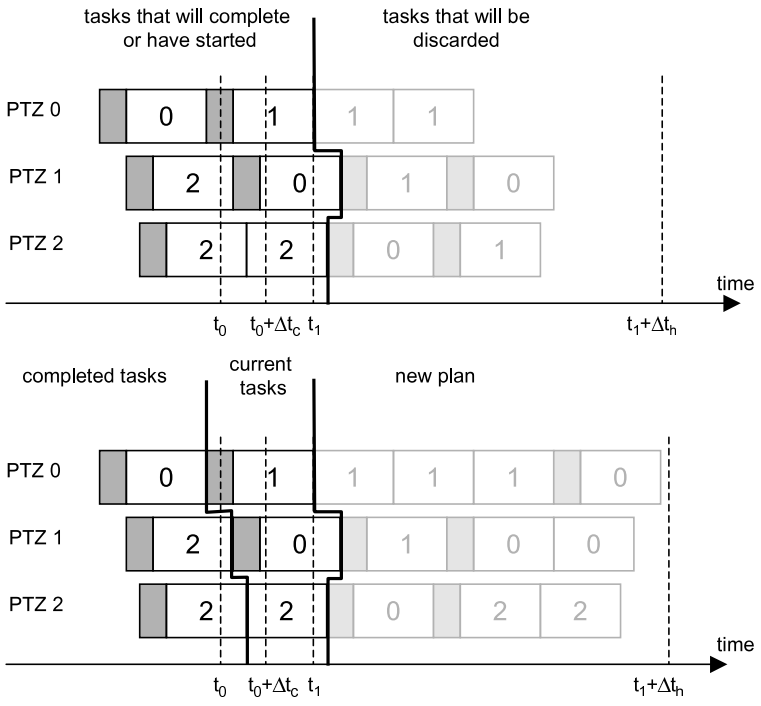


Figure 61: Replacement of PTZ Plans. Top: At time t_0 the system begins to compute a new PTZ strategy, where the optimization is predicted to complete at time $t_0 + \Delta t_c$. All tasks that are executed at that time will be allowed to complete, which is what the optimizer will take as a starting state. All later tasks will be discarded. Bottom: The discarded plans are replaced with an updated plan. The line $t_0 + \Delta t_h$ denotes the plan horizon. Gray boxes denote intervals during which a camera moves from one target to another. White boxes with numbers denote the target ID that is scheduled to be captured.

uled task completes, t_1, plus an offset Δt_h, which is chosen such that at least a certain number of tasks can be assigned to every PTZ camera (see Figure 61).

D.3.2 Combinatorial Search

Finding a plan that fits into the plan horizon while maximizing the objective function (11) is a combinatorial problem. During optimization we iteratively add target to camera assignments to the plan, which implicitly defines a (directed acyclic) weighted graph, where the weights represent changes in quality incurred by the additional assignments, and nodes describe the feasible plans explored so far. It is easy to see in Eq. 11 that the plan quality is non-decreasing in the number of target to plan assignments. Any plan that can not be expanded further without violating the plan horizon is a terminal node in this graph and is a candidate solution. We can utilize standard graph search algorithms such as best first search to find these terminal candidate solutions. Finding the optimal solution requires exhaustively examining this expanding graph, which is possible if the number of observed targets is small. Solving larger problems through approaches such as A* or branch-and-bound searches requires an admissible optimistic estimator that can hypothesize the quality of a non-terminal node during graph expansion. Such a heuristic, however, has so far not been found. We currently resolve to a best first strategy, which rapidly yields good candidate solutions, followed by coordinate ascend optimization through assignment changes. It rapidly improves candidate solutions by making small changes in the assignments in the currently found plan. The search is continued until a preset computation time is exceeded, upon which the algorithm returns the best solution found so far. See algorithm 1 for an overview.

At every optimization step and during PTZ control, the algorithm is provided with data from the site-wide tracking system. If no targets are currently observed, the system could go into idle mode and remain in its home position. However, we employ a slightly more predictive approach. During operation we continuously cluster target arrival locations into a set of multiple clusters [40]. If no active targets are currently present, the system is provided with virtual target locations corresponding to the cluster centers. This essentially leads to the system to adaptively assume a

ready position such that the PTZ cameras are aimed at typical target arrival locations. Since virtual targets are treated just like other target, the system automatically optimizes the PTZ camera to virtual target assignments, giving optimal views at possible arrivals.

Data : Current time t_k. Predicted algorithm completion time Δt_c. Current PTZ activities up to time $t_k + \Delta t_c$. Plan horizon $t_h := t_k + \Delta t_h$. All captures c_{ij}^o that have been (or will have been made) after all current PTZ activities complete.

Result : Optimal plan $\mathcal{E}^* = [\mathbf{E}_1, \ldots, \mathbf{E}_{\mathbf{N}^{\mathrm{P}}}]$ such that objective (11) is maximized.

begin

Set $\mathcal{E}^* = [[], \ldots, []]$.

Compute the partial failure probability with all past captures $F_i = \prod_j (1 - p(S|c_{ij}^o, t_h)$.

Set $p(S|\mathcal{E}^*) = \frac{1}{N} \sum_i [1 - F_i]$.

Create priority queue Q^{open} and insert $(\mathcal{E}^*, p(S|\mathcal{E}^*))$.

Create empty set L^{closed}.

while Q^{open} *not empty and run-time is less than* Δt_c **do**

Retrieve best node $\mathcal{E} = [\mathbf{E}_1, \ldots, \mathbf{E}_{\mathbf{N}^{\mathrm{P}}}]$ from Q^{open} and add it to L^{closed}.

if $\mathcal{E}$ *is a terminal node and* $p(S|\mathcal{E}) > p(S|\mathcal{E}^*)$ **then**

Locally refine plan $\mathcal{E}$ by switching tasks labels until no further increase in quality possible.

Set $\mathcal{E}^* := \mathcal{E}$

Continue.

for *all PTZ cameras* $\mathbf{C}_i^{\mathrm{p}}$ *and all targets* s_l^k, $l = 1, \ldots, \mathbf{N}^t$ *that are active at time* t_k **do**

if *adding* s_l^k *to plan* $\mathbf{E}_i$ *exceeds time limit* t_h **then**

Continue

Create new plan $\mathcal{E}'$ from $\mathcal{E}$ by adding target s_l^k to $\mathbf{E}_i$.

if *not* $\mathcal{E}' \in L^{\mathrm{closed}}$ **then**

Predict the parameters for this new capture c_l and compute quality q_l.

Assume that this plan leads overall to captures $\{c_{xy}\}$ for targets indexed by x and the number of captures (of target x) indexed by y.

Compute $p(S|\mathcal{E}') = \frac{1}{N} \sum_x \left[1 - F_x \prod_y (1 - p(S|c_{xy}^o, t_h)\right]$.

Insert $(\mathcal{E}', p(S|\mathcal{E}'))$ into Q^{open}.

end

The plan $\mathcal{E}^*$ is our best solution.

Algorithm 1: Estimating best PTZ plan.. A best first graph search is used to obtain the best solution within computational time window Δt_c.

E Techinical Details of Social Network Analysis

E.1 Building Social Network

A social network graph, $\mathbf{G} = (\mathbf{V}, \mathbf{E})$, consists of a set of nodes, $\mathbf{V}$, and a set of edges, $\mathbf{E}$. Each node represents an individual in the society and is assigned a name and signature (typically a face image and other identifying information). In a closed-world environment, the number of nodes, N, equals the number of signatures on a given watchlist. To construct $\mathbf{G}$, we estimate the social connection strength, A_{ij}, between two individuals, i and j, based on the following guidelines:

1. Both individual i and j need to be positively recognized.

2. The interaction between them needs to be quantified with a suitable metric.

3. The frequency with which they are seen together should be measured appropriately over time, i.e., we need to aggregate the knowledge gained from the above two guidelines over the "lifespan" of a given set of tracks and recognized faces.

To achieve these guidelines, we begin by leveraging a state-of-the-art face recognition engine [23] that opportunistically, hence often intermittently, recognizes faces detected [24] in the captured images. We obtain from the face recognition engine a discrete probabilistic histogram, $\mathbf{p} = [p^1, \ldots, p^N]$, where each histogram bin, p^i, measures the probability that the recognized face corresponds to individual i. The index, i', of the bin with the highest value is thus, in a probabilistic sense, the ID of the individual. Given a pair of histograms, $(\mathbf{p}, \mathbf{q})$, and $i' = \arg\max_i p^i, j' = \arg\max_j q^j$, we can then update the social connection link, $A_{i'j'}$, with the degree of interactions between individual i' and j', which we model as the motion similarity of i' and j'.

Suppose now that we are given a total of M tracks. We denote each track $m \in M$ by

$$\mathbf{X}_m = \{\mathbf{x}_m^{t_{m,0}}, \ldots, \mathbf{x}_m^{t_{m,\tau}}\}, \tag{14}$$

where $\mathbf{x}_m^t$ is the 3D ground plane location at time t, and $t_{m,0}$ and $t_{m,\tau}$ are the start and end time of the track. Given a pair of tracks, $(\mathbf{X}_m, \mathbf{X}_n)$, which overlaps temporally between (t_0^{mn}, t_τ^{mn}), where

$$t_0^{mn} = \max(t_{m,0}, t_{n,0}), \quad t_\tau^{mn} = \min(t_{m,\tau}, t_{n,\tau}), \tag{15}$$

we quantify their motion similarity as

$$D_{mn} = \exp\left(-\frac{\sum_{t=t_0^{mn}}^{t_\tau^{mn}} \|\mathbf{x}_m^t - \mathbf{x}_n^t\|^2}{2\sigma_{loc}^2(t_\tau^{mn} - t_0^{mn})}\right), \tag{16}$$

such that the more consistently two tracks move together, the larger the similarity D_{mn} is. Here, σ_{loc} is a scaling factor that controls the influence of the variations between the tracks' locations.

Based on Eq. 16, we can now define the rule for updating $A_{i'j'}$ for a pair of recognized faces, $(\mathbf{p}, \mathbf{q})$, and their tracks, $(\mathbf{X}_m, \mathbf{X}_n)$ (finding the association between faces and tracks will be addressed in Section E.1.1), as

$$A_{i'j'} = A_{i'j'} + D_{mn}(\exp^{-\alpha\mathcal{H}(\mathbf{p})} p^{i'} + \exp^{-\alpha\mathcal{H}(\mathbf{q})} q^{j'}), \tag{17}$$

where noisy recognition is elegantly mitigated by the entropy measure $\mathcal{H}(.)$. Here, for a histogram $\mathbf{p}$, the larger $\mathcal{H}(\mathbf{p})$ is, or equivalently, the more uniformly distributed histogram $\mathbf{p}$ is, which indicates ambiguous recognition, the smaller $\exp^{-\alpha\mathcal{H}(\mathbf{p})}$ would be and hence the lesser the influence on $A_{i'j'}$. In addition, Eq. (17) shows that the links are being continuously updated with all valid pairs of faces and corresponding tracks. Hence, the more frequently the system has "seen" the individuals together, the stronger a link is.

E.1.1 Face-to-Track Association via Graph-Cut

So far, we have assumed that the track associated with a recognized face is readily available. This assumption is frequently violated due to the following reasons. During a face capture, the images that are acquired from a PTZ camera could capture one or more faces in different parts of the image. As a result, detected faces have to be projected into 3D space (recall that our tracker operates in 3D

space) in order to be associated with the tracks. Such a projection requires estimating the projection matrix of the PTZ camera as it moves, which, depending on the accuracy of the PTZ motor location provided by the system, is often inaccurate. The situation, where several faces might be detected within a single PTZ view at the same time, also makes it difficult to associate tracks using a simple distance metric due to the proximity of these individuals. Furthermore, these faces must clearly belong to different tracks, a fact that needs to be taken into consideration during track association. On the other hand, faces from different PTZ views could belong to the same individual and should not be used to update the network. See Figure 62 for an illustration. In this section, we will look at how we can build a social network that would still be a realistic representation of the true social interactions between individuals, in spite of these challenges.

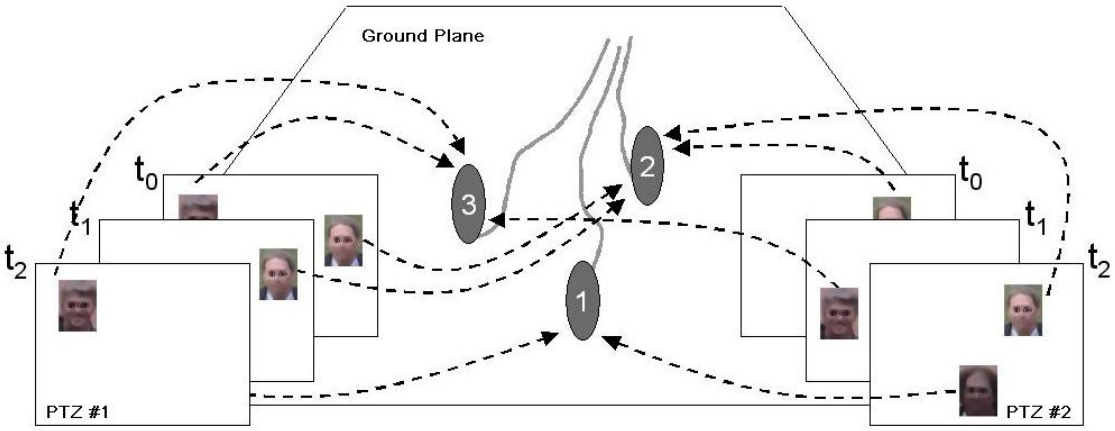


Figure 62: Face-to-Track Association: Three persons are tracked on the ground plane with track ID 1, 2, and 3. Several face images are detected in two different PTZ camera views. The algorithm needs to find the correct correspondence between faces and tracks.

Let us denote a set of R detected faces by $\mathbf{F} = \{\mathbf{f}_1, \ldots, \mathbf{f}_R\}$, and each capture $\mathbf{f}_r$, $r \in R$, contains

$$\mathbf{f}_r = (\mathbf{x}_r, \mathbf{\Sigma}_r, \mathbf{p}_r, t_r, c_r), \tag{18}$$

where t_r is the time of capture, c_r is the index of the PTZ camera that performs the capture, $\mathbf{x}_r$ is the 3D ground plane location of the face computed by backprojecting the detected 2D face location using the estimated projection matrix, $\mathbf{Pm}_r$, of c_r at t_r, $\mathbf{\Sigma}_r$ is the backprojection variance

due to errors in the face location and noisy projection matrix estimation, and finally $\mathbf{p}_r$ is the face recognition histogram as presented in Section E.1. Further let the set of M tracks be $\mathbf{X} = \{\mathbf{X}_1, \ldots, \mathbf{X}_M\}$, where each track $\mathbf{X}_m$ is defined as in Eq. 14. The association problem is then to assign a label l to $\mathbf{f}_r$ so that $l_r \in \{0, 1, \ldots, M\}$ indicates the track this capture belongs to. An extra label 0 is introduced to take care of outlier situations, such as missing tracks and/or face captures that are false positives.

Given such a labeling problem, and the difficulties in associating faces to tracks as mentioned, we propose a Markov Random Field (MRF) framework, and solve it using the multi-way graph-cut algorithm [32, 33, 34]. In our MRF formulation, we define the site set over all face captures $\mathbf{F}$ with $|\mathbf{F}| = R$, and the label set over $\mathbf{X}$ with $|\mathbf{X}| = M + 1$ (after adding the missing track with label $l = 0$). In this framework, we seek to find an optimal labeling, $\mathbf{L}^* = (l_1^*, \ldots, l_R^*)$, where $l_r^* \in \{0, 1, \ldots, M\}$, for all sites by minimizing the following energy function

$$E(\mathbf{L}) = \sum_{r \in R} D(l_r) + \sum_{r,s \in \mathcal{N}} V_{r,s}(l_r, l_s), \tag{19}$$

where the data term, $D(l_r)$, is for evaluating the cost of assigning the face capture $\mathbf{f}_r$ (site r) to track $\mathbf{X}_{l_r}$ (label l_r), and the pairwise smoothness term, $V_{r,s}(l_r, l_s)$, is for computing the cost of assigning the sites (r, s) (face captures $(\mathbf{f}_r, \mathbf{f}_s)$) to labels (l_r, l_s), and $\mathcal{N}$ specifies some neighborhood system.

In order to properly manage detected faces from multiple PTZ views, $\mathcal{N}$ consists of three edge types: (1) the edge between a pair of faces if they are captured from the same camera view at the same time, denoted as $\mathcal{N}_1$, (2) the edge between a pair of faces if they are captured from the same camera view but at two successive time slots, denoted as $\mathcal{N}_2$, and (3) the edge between a pair of faces if they are captured from two different camera views at the same time, denoted as $\mathcal{N}_3$.

For our data term, $D(l_r)$, we simply adopt the strategy that if the 3D face location is closer to one of the track locations than others at the capture time t_r, this face would be more likely to be assigned to this track. Thus,

1. For $l_r = 0$, i.e., the face capture is assigned to a null (missing) track

$$D(l_r) = \delta. \tag{20}$$

2. For $l_r \neq 0$, we have

$$D(l_r) = \begin{cases} d_m(\mathbf{x}_{l_r}^{t_r}; \mathbf{x}_r, \mathbf{\Sigma}_r), & \text{if } t_r \in (t_{l_r,0}, t_{l_r,\tau}) \\ \infty, & \text{otherwise} \end{cases} \tag{21}$$

where $\mathbf{x}_{l_r}^{t_r}$ is the estimated location of track $\mathbf{X}_{l_r}$ at time t_r, $(t_{l_r,0}, t_{l_r,\tau})$ defines the lifespan of this track, $d_m(\sim)$ is the Mahalanobis distance defined by

$$d_m(\mathbf{x}; \mu, \mathbf{\Sigma}) = \sqrt{(\mathbf{x} - \mu)^\tau \mathbf{\Sigma}^{-1} (\mathbf{x} - \mu)}, \tag{22}$$

and δ is set to be some penalty cost for assigning a face to a null track.

Given that the neighboring edges might be of types $\{\mathcal{N}_1, \mathcal{N}_2, \mathcal{N}_3\}$, the smoothness term is defined accordingly as

1. For $(r, s) \in \mathcal{N}_1$, $V_{r,s}(l_r, l_s) =$

$$\begin{cases} -\infty, & \text{if } l_r \neq l_s \\ 0. & \text{otherwise} \end{cases} \tag{23}$$

2. For $(r, s) \in \{\mathcal{N}_2, \mathcal{N}_3\}$, $V_{r,s}(l_r, l_s) =$

$$\begin{cases} -0.5 \exp\{-\beta d_b(\mathbf{p}_r, \mathbf{p}_s)\} * \\ (d_m(\mathbf{x}_r; \mathbf{x}_s, \mathbf{\Sigma}_s) + d_m(\mathbf{x}_s; \mathbf{x}_r, \mathbf{\Sigma}_r)), & \text{if } l_r \neq l_s \\ 0, & \text{otherwise} \end{cases} \tag{24}$$

where $d_b(\mathbf{p}_r, \mathbf{p}_s)$ is the Bhattacharyya coefficient of two histograms defined as

$$d_b(\mathbf{p}_r, \mathbf{p}_s) = \sum_{i=1}^{N} \sqrt{p_r^i p_s^i}. \tag{25}$$

The idea here is that for case $\mathcal{N}_1$, if they are correctly assigned to two different tracks, there would be a tremendous payoff of $-\infty$. For cases $\mathcal{N}_2$ and $\mathcal{N}_3$, the payoff for assigning two faces to different tracks depends on the Mahalanobis distances between $\mathbf{x}_r$ and $\mathbf{x}_s$, and the similarity between their face recognition histograms evaluated by the Bhattacharyya coefficient. The more distant (in space) the faces are from each other and the more dissimilar their face recognition histograms are, the larger the payoff.

Finally, as mentioned, we could solve for Eq. 19 using the multi-way graph-cut algorithm, which can efficiently generate a solution within a known factor of the optimal. The resulting face-to-track associations can then be utilized for updating the social links as presented in Section E.1.

E.2 Discovering Community Structure via Modularity-Cut

After building a social network, we follow on to determine its *community structure*, i.e., the social groups that it might contain. A social group, in our case, is defined as a cohesive group of individuals that are frequently seen together. Individuals in the same social group display strong connections between one another in the social network. From a graph-theoretic perspective, the problem is to divide the social network into subgraphs in a way that maximizes connections between nodes in each subgraph and minimizes connections between different subgraphs. A closer look at the problem should reveal the applicability of several well-known spectral clustering techniques [35, 2, 36, 37]. Most of these techniques approach the problem by looking for divisions that minimize the connections between subgraphs, or in other words, divisions that minimize the *cut size* [36].

E.2.1 Dividing into Two Social Groups

The utility of the above techniques for discovering community structure is limited by the requirement to know the number of social groups beforehand, which is impractical for our purpose. Shi et al. [2] have proposed that one can recursively divide the graph into two based on the normalized-cut. We take a similar recursive approach but instead of using cut size as the criterion, we adopt

an approach originally proposed by Newman [22, 38] in the domain of social network study. He argued that using cut size as the division criterion is counter-intuitive to the concept of social group and that one instead needs to maximize the *modularity measure* [41], which expresses the difference between the actual and expected connections of individuals within each social group. We call this technique the modularity-cut, which has been shown [22, 38, 41] in many real-world examples to be superior at identifying social groups.

To understand the modularity-cut, one can consider the notion that two individuals, i and j, are strongly connected only if their connection is stronger than what would be *expected* between any pair of individuals, that is,

$$B_{ij} = A_{ij} - \frac{k_i k_j}{2m}, \tag{26}$$

where A_{ij} is the connection strength between i and j, k_i and k_j are the total connection strengths of i and j (i.e., $k_i = \sum_j A_{ij}$), and $m = \frac{1}{2}\sum_{ij} A_{ij}$ is the total strength of all connections in the social network graph. The term $\frac{k_i k_j}{2m}$ represents the expected edge strength, so that the further an edge (A_{ij}) deviates from expectation, the stronger the connection. From Eq. 26, the *modularity measure*, Q, can be easily derived as

$$Q = \frac{1}{2m}\sum_{\substack{i,j \in \\ \text{same} \\ \text{group}}} B_{ij} = \frac{1}{4m}\mathbf{s}^T\mathbf{B}\mathbf{s}, \tag{27}$$

where $\mathbf{s}$ is a labeling vector with each element, s_i, corresponding to an individual (node) in the social network graph. $s_i = +1$ if node i is assigned to the first group and $s_i = -1$ if node i is assigned to the second. $\mathbf{B}$ is the modularity matrix whose elements are B_{ij}. Thus, each time we divide a graph into two subgraphs, as opposed to "simply" minimizing cut size, we maximize modularity Q using $\mathbf{B}$. This is an appealing notion - by maximizing Q we favor stronger than expected within-group connections and weaker than expected between-group connections (the cut).

Determining $\mathbf{s}$ that maximizes Q can be shown to be NP-hard. We will next outline a method based on Eigen-analysis that will provide a good approximation [38] to our problem. We first

perform an eigen decomposition $\mathbf{B} = \sum_i \beta_i \mathbf{u}_i {\mathbf{u}_i}^T$ with eigenvalues β_i and eigenvectors $\mathbf{u}_i$. Substituting into Eq. (27) we obtain

$$Q = \frac{1}{4m} \sum_i ({\mathbf{u}_i}^T \mathbf{s})^2 \beta_i. \tag{28}$$

Several key observations can be made in regards to Eq. 28:

1. If we let $\mathbf{s} = \mathbf{u}_i$, then since the eigenvectors are orthogonal, ${\mathbf{u}_{j|j\neq i}}^T \mathbf{s} = 0$.

2. Since $\mathbf{s}$ is constrained to be ± 1, $\mathbf{s}$ cannot be directly assigned to an eigenvector, which is real valued. Otherwise Q could be maximized by setting $\mathbf{s}$ equal to the dominant eigenvector, $\mathbf{u}_{\max}$.

3. We can, however, assign s_i to $+1$ if the corresponding element in the dominant eigenvector is positive, and -1 otherwise, i.e.,

$$s_i = \begin{cases} +1 & (\mathbf{u}_{\max})_i \geq 0, \\ -1 & (\mathbf{u}_{\max})_i < 0, \end{cases} \tag{29}$$

   where $(\mathbf{u}_{\max})_i$ is the i^{th} element of $\mathbf{u}_{\max}$. In doing so, we make an assumption that $\mathbf{s}$ remains close to being orthogonal to the other eigenvectors so that majority of the mass of the summation will come from the largest eigenvalue, thereby giving us the largest Q. It has been shown [38] that this assumption holds well in practice.

4. If none of the eigenvalues are positive, it implies that based on the modularity measure, there should be no division. This is a desirable behavior.

5. Classical spectral clustering techniques, which minimize the cut size, have to deal with the trivial case of zero cut size whereby all the nodes in the social network graph are placed in one group. In contrast, since we are *maximizing* the modularity, there is no such problem.

E.2.2 Dividing into Multiple Social Groups

The strategy for dividing a graph into two subgraphs can be naturally extended to finding multiple social groups by applying the modularity-cut recursively to each subgraph. For this purpose, it is important to point out that it is possible for an element in $\mathbf{u}_{\max}$ to have a value extremely close to zero. In such cases, regardless of the signs of the elements, they should be assigned to the same subgraph. This is because by being ≈ 0, these elements do not belong to either groups (Eq. 29), and should be kept together just in case subsequent divisions determine that they belong to the same group.

To ensure that the contributions to the modularity measure generated by subsequent divisions is correctly computed, it is imperative that such contributions be related to the original graph. This is different from the recursive approach proposed in Shi et al. [2], whereby each subgraph is treated independently. In our case, this amounts to removing the edges to other subgraphs and results in maximizing the wrong modularity measure.

We first define a $n \times c$ community structure matrix, $\mathbf{S}$, where n is the number of nodes in the social network graph and c is the number of social groups. We begin with $c = 1$, i.e., there is only one group (the entire social network graph), but c increases as we recursively divide into multiple groups. The $(i, j)^{th}$ element of $\mathbf{S}$ is 1 if node i belongs to social group j, and 0 otherwise. It is obvious that the modularity can be equivalently measured as

$$Q = \mathbf{Tr}(\mathbf{S}^T \mathbf{B} \mathbf{S}), \tag{30}$$

where $\mathbf{Tr}$ represents the trace operator, and $\mathbf{B}$ is the original modularity matrix. Based on Eq. 30, the strategy for dividing into multiple social groups is as follow. Each time we obtain a new social group as described in Section E.2.1, we generate a new community structure matrix, $\mathbf{S}'$, with an additional column corresponding to the new group. Denoting the modularity for $\mathbf{S}'$ as Q' and the largest Q in the recursion so far as Q_{max}, the contribution, ΔQ, to the modularity measure is

simply

$$\Delta Q = Q' - Q_{max}, \tag{31}$$

such that if $\Delta Q \leq 0$, the new group is "discarded". The algorithm at this point is very satisfying. It is much simpler to check for zero or negative contribution to the modularity before terminating the division process than, for example, using a pre-specified cut size as the termination condition.

E.2.3 Eigen-Leaders

Furthermore, based on Eq. 28, the modularity-cut approach provides a simple way for identifying the leader of a social group. That is, the leader, ℓ, of a social group is simply

$$\ell = \arg\max_i (\mathbf{u}_{\max})_i. \tag{32}$$

The rationale here is simple - elements of the dominant eigenvector with large magnitudes make large contributions to the modularity.

Let us consider the social network graph, which is constructed on the basis of the frequency with which individuals are seen together. The leader of a social group, G, can be thought of as the individual, ℓ, in the group that was seen, on average, most frequently with everyone else in the same group. The value of $B_{\ell j}$ in Eq. 26, where $j \in G$, would be the highest among all possible $B_{ij|i,j\in G}$. Consequently, the corresponding element in $\mathbf{u}_{\max}$ would have the largest magnitude among its group members. On the other hand, since spectral clustering techniques minimize the cut size, there is no intuitive manner for exploiting the eigenvectors to identify a group leader.

References

[1] Edward Rosten and Tom Drummond. Fusing points and lines for high performance tracking. In *ICCV '05: Proceedings of the Tenth IEEE International Conference on Computer Vision*, pages 1508–1515, Washington, DC, USA, 2005. IEEE Computer Society.

[2] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, August 2000.

[3] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*. The MIT Press, 2nd edition, 2001.

[4] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proc. International Joint Conference on Artificial Intelligence*, 1981.

[5] Jianbo Shi and Carlo Tomasi. Good features to track. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, Seattle, Washington, June 1994.

[6] C. Tomasi and T. Kanade. Detection and tracking of point features. Technical Report CMU-CS-91-132, Carnegie Mellon University, April 1991.

[7] L. Marchesotti, L. Marcenaro, and C. Regazzoni. Dual camera system for face detection in unconstrained environments. In *IEEE Conf. Image Processing*, volume 1, pages 681–684, 2003.

[8] C. Stauffer and W.E.L. Grimson. Learning patterns of activity using real-time tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):747–757, August 2000.

[9] Tao Zhao and R. Nevatia. Tracking multiple humans in complex situations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(9):1208– 1221, September 2004.

[10] N. Krahnstoever, P. Tu, T. Sebastian, A. Perera, and R. Collins. Multi-view detection and tracking of travelers and luggage in mass transit environments. In *In Proc. Ninth IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETS)*, 2006.

[11] P. Tu, F. Wheeler, N. Krahnstoever, T. Sebastian, J. Rittscher, X. Liu, A. Perera, and G. Doretto. Surveillance video analytics for large camera networks. In *SPIE Newsroom*, 2007.

[12] J. Krumm, S. Harris, B. Meyers, B. Brumitt, M. Hale, and S. Shafer. Multi-camera multi-person tracking for easyliving. In *IEEE Workshop on Visual Surveillance*, July 2000.

[13] Robert Collins, Alan Lipton, and Takeo Kanade. A system for video surveillance and monitoring. In *Proceedings of the American Nuclear Society (ANS) Eighth International Topical Meetingon Robotics and Remote Systems*, April 1999.

[14] N. Krahnstoever and P. Mendonça. Bayesian autocalibration for surveillance. In *Proc. of IEEE International Conference on Computer Vision (ICCV'05), Beijing, China*, October 2005.

[15] K. A. Tarabanis, P. K. Allen, and R. Y. Tsai. A survey of sensor planning in computer vision. *IEEE Transactions on Robotics and Automation*, 11(1):86–104, 1995.

[16] X. Zhou, R. Collins, T. Kanade, and P. Metes. A master-slave system to acquire biometric imagery of humans at a distance. In *ACM SIGMM Workshop on Video Surveillance*, pages 113–120, 2003.

[17] A. Hampapur, S. Pankanti, A. Senior, Y. L. Tian, and L. B. Brown. Face cataloger: Multi-scale imaging for relating identity to location. In *Proceedings of the IEEE Conference on Advanced Video and Signal Based Surveillance, Washington D.C.*, pages 13–21, 2003.

[18] F. Qureshi and D. Terzopoulos. Surveillance camera scheduling: A virtual vision approach. *ACM Multimedia Systems Journal, Special Issue on Multimedia Surveillance Systems*, 12(3):269–283, 2006.

[19] A. D. Bimbo and F. Pernici. Towards on-line saccade planning for high-resolution image sensing. *Pattern Recognition Letters*, 27(15):1826–1834, 2006.

[20] S. N. Lim, L. S. Davis, and A. Mittal. Constructing task visibility intervals for a surveillance system. *ACM Multimedia Systems Journal, Special Issue on Multimedia Surveillance Systems*, 12(3), 2006.

[21] S. N. Lim, L. Davis, and A. Mittal. Task scheduling in large camera network. In *Proceedings of the Asian Conf. on Computer Vision, Tokyo, Japan*, 2007.

[22] MEJ Newman. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences*, 103(23):8577–8582, 2006.

[23] Identix. Faceit sdk.

[24] H. Schneiderman and T. Kanade. A statistical method for 3D object detection applied to faces and cars. In *IEEE Computer Vision and Pattern Recognition, Hilton Head, SC*, volume 1, pages 746–751, 2000.

[25] T. Yu, Y. Wu, N. O. Krahnstoever, and P. H. Tu. Distributed data association and filtering for multiple target tracking. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, Alaska*, June 2008.

[26] Samuel Blackman and Robert Popoli. *Design and Analysis of Modern Tracking Systems*. Artech House Publishers, 1999.

[27] C. Rasmussen and G. Hager. Joint probabilistic techniques for tracking multi-part objects. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 16–21, 1998.

[28] Nils Krahnstoever, Ting Yu, Ser-Nam Lim, and Kedar Patwardhan. Multiclass spectral clustering. In *Workshop on Multi-camera and Multi-modal Sensor Fusion, in conjunction with ECCV 2008*, 2008.

[29] B. Bose, Xiaogang Wang, and E. Grimson. Multi-class object tracking algorithm that handles fragmentation and grouping. In *Computer Vision and Pattern Recognition*, pages 1–8, jun 2007.

[30] A.P. French, A. Naeem, I.L. Dryden, and T.P. Pridmore. Using social effects to guide tracking in complex scenes. In *Advanced Video and Signal Based Surveillance*, pages 212–217, 2007.

[31] G. Gennari and G.D. Hager. Probabilistic data association methods in visual tracking of groups. In *Computer Vision and Pattern Recognition*, volume 2, pages 876–881, jun 2004.

[32] Yuri Boykov and Vladimir Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE transactions on PAMI*, 26(9):1124–1137, 2004.

[33] Yuri Boykov, Olga Veksler, and Ramin Zabih. Efficient approximate energy minimization via graph cuts. *IEEE transactions on PAMI*, 20(12):1222–1239, 2001.

[34] Vladimir Kolmogorov and Ramin Zabih. What energy functions can be minimized via graph cuts? *IEEE transactions on PAMI*, 26(2):147–159, 2004.

[35] Andrew Y. Ng, Michael I. Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems 14*, pages 849–856. MIT Press, 2001.

[36] Ulrike von Luxburg. A tutorial on spectral clustering. *CoRR*, abs/0711.0189, 2007.

[37] Stella X. Yu and Jianbo Shi. Multiclass spectral clustering. In *International Conference on Computer Vision*, pages 313–319, 2003.

[38] M. E. J. Newman. Finding community structure in networks using the eigenvectors of matrices. *Physical Review E*, 74:036104, 2006.

[39] N. Krahnstoever and P. Mendonça. Autocalibration from tracks of walking people. In *Proc. British Machine Vision Conference (BMVC), Edinburgh, UK, 4-7 September*, 2006.

[40] C. Stauffer. Estimating tracking sources and sinks. In *Proceedings of the Second IEEE Workshop on Event Mining*, July 2003.

[41] M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Phys. Rev. E*, 69, 2004.