GE
Global Research

Advanced Behavior Recognition in Crowded Environments

Ming-Ching Chang
Weina Ge
Nils Krahnstoever
Ting Yu
Ser Nam Lim
Xiaoming Liu (PI)

Final Report

Sensor Surveillance Program
Grant Number 2009-SQ-B9-K013
Reporting period: October 2009 to September 2011

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

September 30, 2011

imagination at work

Table of Contents

1 Abstract

This document is the final report for the NIJ research program "Advanced Behavior Recognition in Crowded Environments". The goal of this program is to increase the situational awareness in law-enforcement and correctional settings and reliably detect and prevent activities indicative of disorderly conduct and criminal behavior. Examples include fights, riots, the formation of drug markets, and gang activities. A particular emphasis of this program is to develop robust probabilistic event modeling framework that takes the uncertainty of low-level image evidence into consideration. In addition, our technology aims for user friendly interaction to the law-enforcement end users by developing event explanation and scenario modeling GUI.

Some of the accomplishments of this program are: (i) a resource description framework (RDF) that is responsible for dynamically representing and maintaining probabilistic and non-probabilistic meta data is developed, which plants the foundation for the following probabilistic event recognition and learning-based event recognition; (ii) a probabilistic event recognition system combining low-level probabilistic evidence and rule-based domain knowledge has been developed that enables the detection of pre-defined events from either video achieves or real-time video feeds; (iv) features such as event explanation and scenario modeling GUI are implemented to increase the usability of our system for law-enforcement end users; (v) a novel framework for learning-based event recognition is developed that can achieve satisfying recognition in real-time processing; (vi) the system was tested live during the 2010 Mock Prison Riot sponsored by the NIJ as well as evaluated against real-world video data that was collected from the surveillance camera network at Schenectady NY.

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 law enforcement detect many different types of events 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.

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

2 Executive Summary

This progress report covers the time period from October 2009 through September 2011. In the following, an executive summary of the extended report is provided.

2.1 Data Collection

During this program, video data were drawn from three sources.

I. Continuing from the success of the 2009 Mock Prison Riot from our previous NIJ program, we perform data collection and system demonstration at the 2010 Mock Prison Riot sponsored by the NIJ. During this very successful event, more than 15 hours of high-quality video material was captured from four cameras including an optional IR camera, amounting to more than $80GB$ of video material. See Figure 1 for an overview of the datasets.

II. In our collaboration with the Schenectady City Police Department (SCPD), we have access to a large amount of real-life security videos from their city-wide security camera network, the Public Surveillance Camera Project (PSCP). The availability of surveillance videos of crime scenarios including drug-dealing and stubbing murder greatly enriches the development of video analytic algorithms for real-life applications.

III. We have also collected videos of moving crowds and their facial shots using multiple cameras and Pan-Tilt-Zoom (PTZ) cameras in order to perform gaze and social interaction analysis in the GE GRC Courtyard. Volunteers are recruited from the general population of GE employees. This dataset focused on the motion patterns and interaction of people in analyzing social behaviors.

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

2.2 Intelligent Video – Group-Level Scenario Recognition

GE Global Research's comprehensive Intelligent Video platform is expanding continuously through separately funded GE research projects. The core of the platform is a robust and scalable surveil-

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

Figure 1: **Datasets.** Three examples of datasets utilized as part of this program. (a) Data collected at the 2010 NIJ Mock Prison Riot. (b) Data collected at a city-wide security camera network, the Public Surveillance Camera Project (PSCP) in our collaboration with a local police department, the Schenectady City Police Department (SCPD) for deployment and validation. (c) Data collected at the GE Global Research Center (GRC) to study PTZ facial shots and gaze tracking.

lance system with multi-view, multi-target tracking capability. The system is able to track subjects across large camera networks. The system operates in real-time and is able to handle challenging and crowded conditions. It has been successfully deployed at various locations for validation and demonstration, including a sports arena for the DHS, an airport, and in retail environments where it has helped operators to prevent and detect shop-lifting.

This program is built on the aforementioned platform to perform group-level event recognition and motion pattern analysis for surveillance. Specifically, we address the key question of continuous automated video-based behavior recognition in public venues where continuous law enforcement presence is desirable but infeasible. Our system has two components:

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

- a *probabilistic group analysis* to reason over the soft group structure between individuals based on a connectivity graph defined using track-to-track and path-based connectivity measures.
- a *probabilistic motion analysis and scenario recognition module* to reason over spatial-temporal patterns and interactions, both at individual and group levels.

Our approach is capable of handling arbitrary number of participants. The generic group representation can be combined naturally with subsequent probabilistic behavior reasoning. Analytical event rules can be derived directly by combining individual probabilistic inference modules. Our recognition framework is thus flexible in adapting to new scenarios. Moreover, our model construction is intuitive and tractable for non-technical users. It is also invariant to site-specific observations. The main technical contributions of this program include:

- a probabilistic group/crowd analysis module which serves as the baseline for event recognition
- resource description, modular inference, and verbal explanation of triggered events
- scenario modeling with an operator-friendly GUI frontend
- biometrics (face/gaze) tracking for group/social network recognition

We summarized these major efforts in details as follows.

Evidence Representation – Resource Description: We have developed a resource description framework (RDF) [1] responsible for dynamically representing and maintaining all knowledge and meta data regarding the entire surveillance system, including all tracking results, inference results, and triggered events. The RDF creates a registry of name-value pairs that provides a hierarchical representation of system meta-data in memory. The design of the RDF is *non-declarative* in the sense that the consumer of meta data does not have to be aware of its content or type. In addition, the RDF is *dynamic* and can be extended by outside components. These properties of the RDF enable the development of modular "plug-in" components that can query the RDF, perform inference, draw conclusions, and then add new information back to the RDF tree. All evidence that is estimated in either a probabilistic or a non-probabilistic fashion can be stored in a flexible,

centralized data structure that is accessible from our system as well as outside components using simple scripts.

Probabilistic Low Level Evidence: We have created a wide range of low-level motion evidence modules, which serve as the building blocks for further analysis in other components:

- individual's motion types: is a person moving fast, slow, or loitering? is one standing, walking, or running? Does the person belong to a crowd?
- relative motion direction and distance change between pairs of people: is the person A approaching or chasing person B? are they going to meet or intercept?

These basic evidence detection modules can be combined to perform advanced, high-level inference.

Probabilistic Group Analysis: Defining a proper *grouping* of a crowd is challenging due to complex social interactions and relations that are hard to measure precisely. We maintain a soft, probabilistic grouping measure in order to handle uncertainties in video tracking, in contrast to other approaches that explicitly defines segments for groups. We first seek an instantaneous pairwise group affinity measure that represents the probability of a pair of people belonging to a group, by checking if they are physically close. Inspired by standard social norms from Hall's *proxemics* theory [2] for modeling inter-person spacial relations and the social force model [3] for modeling pedestrian dynamics, we define a pairwise grouping measure based on three terms: the *distance* between the individuals, their *motion* (body pose and velocity) and the *track history*. We further introduce a path-based group connectivity that estimates the pairwise grouping probability under the influence of others. Specifically, we set the connection probability between individuals to be the optimal path amongst all possible paths, and cast the problem into an all-pair shortest path finding problem. This path-based grouping is shown to be less biased and serves as a main tool for group behavior analysis. A technical paper covering this work will be published in [4].

Automatic Scenario Recognition: Once a probabilistic estimation of a scenario is computed, the system must decide whether or not to trigger an alert to the operator. We follow a standard Receiver Operating Characteristic (ROC) analysis [5, 6] to determine a baseline threshold of the

ideal performance for event triggering. We manage triggered events by keeping track of their duration to avoid repeated triggers of the same event. Specifically, we model event duration by using an *armory mechanism*, that is, once an event is triggered, any subsequent event detected during a period of time is kept silent and only used to update (extend) the period of time. In other words, the same event will be triggered only after it is not detected for some pre-defined duration of time. We found this mechanism effective in filtering out unwanted events in practice.

Event Explanation: Since our inference engine is probabilistic (Bayesian), and is highly modular, the explanation of its triggered events is *explicit* from the reasoning process — the probability in reasoning can provide straightforward verbal explanations. For example, the explanation of a loitering event could be: "The loitering event is detected for target A with $P(loiter) = 0.71$ because: (1) the target is currently moving slow with $P(slow) = 0.86$, (2) the target has been close to its current position at a point in time in the past within a window of 10.0 and 20.0 seconds ago, and (3) the target was moving slow at that previous point in time with $P(slow) = 0.33$." Furthermore, backtracking in explanation is possible. For example, the system can trace back from the loitering probability and further explain how the person was determined to be slow with $P(slow) = 0.33$ in the past.

Scenario Modeling GUI: To enable operators to quickly create models that recognize domain-specific scenarios, we have developed a *visual programming* framework that represents scenarios by a flow of information through a network of processing steps. This approach is motivated by the proliferation of graphical models [7] in general and Bayesian Networks [8] specifically for recognition. In visual programming, algorithms and computational procedures are represented by nodes connected by directional edges. For a given node, its incident edges represent incoming data and exiting edges represent the data produced by this node. Visual programming paradigms have emerged from many different applications such as the programming of toy robots and industrial measurement and simulation systems.

Gaze Tracking: We present a comprehensive approach to track a one or more individuals' gaze angles by estimating their locations, body poses, and head poses in an unconstrained environment.

The approach combines person detections from fixed cameras with directional face detections obtained from actively controlled pan tilt zoom (PTZ) cameras. The main contribution of this work is to estimate both body pose and head pose (gaze) independently from motion direction, using a combination of sequential Monte Carlo Filtering and Markov Chain Monte Carlo (MCMC) sampling. There are numerous benefits in tracking body pose and gaze in surveillance. It allows to track people's focus of attention, can optimize the control of active cameras for biometric face capture, and can provide better interaction metrics between pairs of people. The availability of gaze and face detection information also improves localization and data association for tracking in crowded environments. The performance of the system will be demonstrated on data captured at a real-time surveillance site.

2.3 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 scenario recognition algorithms (Sections 8 and 10) can automatically detect events of interest relevant to law enforcement. These include automatically detecting and predicting suspicious events such as aggression and fighting to keep public parks and correctional facilities safe. Triggered alerts can be used to quickly dispatch officers to the scene of the event and also, if available, aim additional PTZ cameras to obtain higher resolution footage of the event as an evidence.

- The Scenario Modeling GUI (Section 9) is one of the key innovations making our system to be friendly to the end user, law-enforcement practitioners. It allows non-experts to easily define new events of interests, which will automatically enable our system to detect the newly defined event without the assistance from IT experts. This important system property enables law-enforcement practitioners to constantly adapting our system to an ever-changing application domain.

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

- The event explanation system can provide detailed verbal descriptions about detected events to the law-enforcement practitioners to help them better understand the system's capabilities and recognize the potential false alarm from the verbal explanations.

- The gaze and pose estimation system (Section 11) is a spin-off capability developed under this program. The outcome of this algorithm can be used to enable the capture of high-resolution facial shots for pedestrians in crowd environments, which has important applications for law enforcement. This directly enables methods for performing face recognition and face cataloging of uncooperative individuals from a distance.

2.3.1 System Deployment

Given the fact that the earlier versions of our system has been deployed and tested in operator-enacted video data such as Mock Prison Riot, we have been looking for a surveillance site where real-world events and activities take place. To this end, we chose the camera network at Schenectady NY as our deployment and testing site.

Over the years, the Schenectady County District Attorney's Office (DA) and the Schenectady City Police Department (SCPD) have devoted tremendous effort and resources, both public and private, toward implementing the Schenectady's Public Surveillance Camera Project (PSCP). To date, PSCP has expanded its coverage to include areas covered by approximately 72 surveillance cameras, all of which are IP-based PTZ cameras that transfer captured videos to SCPD via a wireless network, covering large sections of downtown Schenectady. After visiting the SCPD and seeing the video data, we concluded that PSCP can be a very good testing site for our event detection system. It provides validation data for our system in a real-world setting, the results of which may provide additional value to DA and SCPD's law enforcement practices and prosecution needs.

We have made a number of visits to SCPD and selected a subset of its video archives that contain the recording of past events that have already been viewed and considered closed. These video data have been transferred to GRC. Our system is able to achieve reasonably good performance on

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

these data.

At this moment, we are working with SCPD to install three additional cameras (2 static cameras and 1 PTZ camera) for PSCP. Once installed, we will deploy a workstation and our software system to SCPD and connect these three cameras directly to our system, which will process the videos in real-time and record the analysis results.

Section 12.2 describes further details on our system deployment and collaboration with the SCPD.

2.3.2 System Evaluation and Feedback

We performed a thorough performance evaluation based on the data collected at the 2009 and 2010 Mock Prison Riot (MPR). From MPR 2009, 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. The general feedback was very positive and the correctional officers where enthusiastic about what the technology is able to accomplish in their operational environments. Details on MPR 2010 evaluation is in Section 12.1.

A recent progress is that we have improved the event recognition performance by at least 10% by leveraging a state-of-art machine learning technique, as shown in a technical paper that we submitted for peer review in Appendix G.

Besides the field testing at the MPR 2009/2010 and the deployment in SCPD, we will also provide support for third party evaluation conducted by ManTech on behave of NIJ.

2.4 Dissemination of Research Results

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

[1] Ming-Ching Chang, Nils Krahnstoever, Sernam Lim, and Ting Yu, “Group Level Activity

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

Recognition in Crowded Environments across Multiple Cameras", In Proc. Seventh IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), Workshop on Activity Monitoring by Multi-Camera Surveillance Systems (AMMCSS), Boston, MA, pp. 56–63, Aug.-Sep., 2010.

[2] Karthik Sankaranarayana, Ming-Ching Chang, and Nils Krahnstoever, "Tracking Gaze Direction from Far-Field Surveillance Cameras", In Proc. IEEE Workshop on Applications of Computer Vision and Applications (WACV), Kona, Hawaii, pp. 519–526, January, 2011.

[3] Nils Krahnstoever, Ming-Ching Chang, and Weina Ge, "Gaze and Body Pose Estimation from a Distance", In Proc. Seventh IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), Klagenfurt, Austria, August, 2011 (Best Paper (Runner Up) Award).

[4] Ming-Ching Chang, Nils Krahnstoever, and Weina Ge, "Probabilistic Group-Level Motion Analysis and Scenario Recognition", In Proc. IEEE 13th International Conference on Computer Vision (ICCV), Barcelona, Spain, Nov., 2011.

The following provisional patent related to part of efforts on gaze tracking is in progress of filing:

Nils Krahnstoever, Peter Tu, Ming-Ching Chang, Weina Ge, "Person Tracking and Interactive Advertising", Provisional Filing, Application Serial No. 13/221,896.

The work covered in this grant has also been featured on the front page of the New York Times (Figure 70).

2.5 Next Steps

In our following up proposal "Advanced Behavior Recognition in Crowded Environments - Continuation" submitted to NIJ on July 2011, we propose to improve the behavior recognition of the system based on the capabilities and motion pattern event detectors developed under this program. First of all, with the goal of being applicable in all surveillance sites, this new system will continue to improve its multi-target tracking component so that it is invariant to various weather-related and lighting conditions. We will design new algorithms for both person tracking and automatic PTZ

control using a single PTZ camera. We will also focus our scenario recognition efforts on complex long duration events using both learning-based approaches and logic reasoning. We will pay particular attention to the system considerations with the goal of reliable operation in crowded real-world environments. The proposed program will draw on video data collected during previous NIJ efforts as well as additional data to be collected from relevant law enforcement locations. In particular, we plan to use the surveillance camera network at the Schenectady City Police Department for data collection, testing, and demonstration of the developed tools.

3 Introduction

In the rest of the report we will present a comprehensive description of our research program. Figure 2 shows the overall structure of our system in relationship to components that are previously developed by GE, third parties or on earlier NIJ programs.

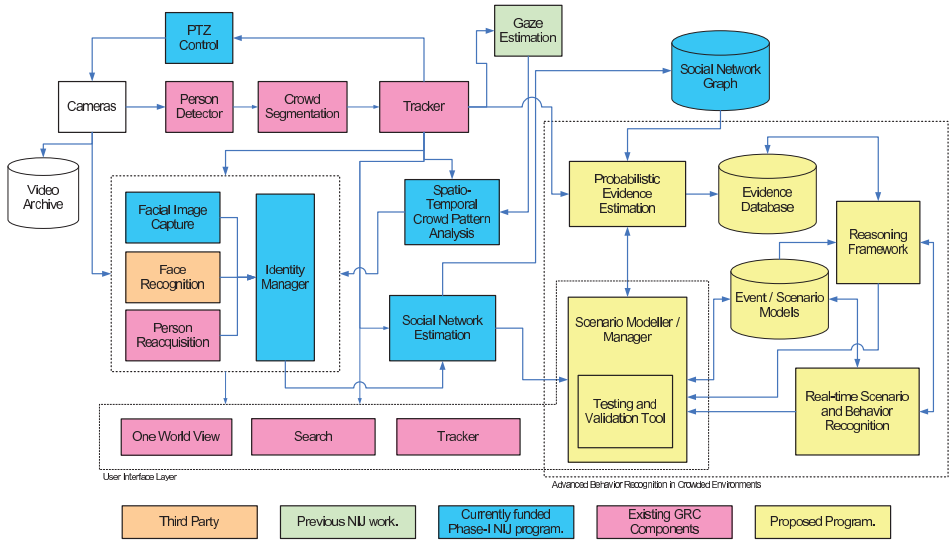


Figure 2: **System Overview:** Components developed in the current program (yellow) in realtionship to previous funded NIJ works (green, blue), GE components (pink), and third parties (orange).

The core track of this report is summarized as follows. We will first give an overview of the data sets that were collected and used, including the NIJ Sponsered Mock Prison Riot event in Chapter 4. In Chapter 5 we will describe our Resource Description Framework (RDF) for evidence representation, where the visual evidence are stored for reasoning modules to effectively perform inference. Chapter 6 describes how low-level evidence can be deduced probabilistically from motion tracking and appearance cues from the video. Based on the low-level evidence collected from individual tracks, Chapter 7 elaborates how we perform group-level probabilistic behavior reasoning. Chapter 8 describes how each scenario of interest can be detected and recognized using the above probabilistic representation. Chapter 9 descirbes the GUI front-end we developed to facilitate customized scenario recognition. Chapter 10 discusses how we extend our scenario

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

recognition approach by leverging recent advanced methods from machine learning and artificial intelligence.

The remaining chapters elaborate topics complement to the core track and complete this report. Chapter 11 describes our effort on biometric (face) detection and tracking, which is closely related to our previous efforts in attention recognition using gaze analysis and social network analysis. Chapter 12 describes progresses in system deployment to law enforcement and system evaluation conducted with NIJ. In the appendices we provide detailed information about the list of all reviews and meetings of this research program. We have also inlcuded all technical papers published as a result of this program.

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 externally or publicly available datasets. Continue from the success of Mock Prison Riot (MPR) event in 2009 from our previous NIJ program, in 2010 we perform additional data collection and testing at the 2010 Mock Prison Riot (MPR) in Moundsville, WV.

4.1 Mock Prison Riot 2010 Data

Based on the recommendation from NIJ we utilized the 2010 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 collection was very successful and we have received high marks from Law Enforcement and Corrections (LEC) practitioners.

About Mock Prison Riot. 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.

4.1.1 Venue

The MPR takes place at the decommissioned West Virginia Penitentiary in Moundsville. It is a four-day, comprehensive LEC tactical and technology experience, including 44,000 square feet of exhibit space, training scenarios, technology demonstrations, technology assessments and evaluations, certification workshops, a Skills Competition, and unlimited opportunities for feedback, networking, and camaraderie. The penitentiary grounds consist of several large cell blocks at-

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

tached to several outdoor recreational yards. Figure 3 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 3: **West Virginia Penitentiary Yards.** (a) aerial view of the prison grounds, (b) recreational yard, and (c) large outdoor yard.

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 3) was chosen as the one central venue for testing and data collection.

4.1.2 Background

Introduction and Relationship Building. During the fall of 2008, members of the GE Global Research Intelligent Video System Team were referred to the MPR by Mr. Jack Harne, Program Manager, and Dr. Frances Scott, Senior Program Manager, both of NIJ's Office of Science and

Technology. The intent was to facilitate formal collaboration between MPR resources and Dr. Nils Krahnstoever, GE Global Research.

Both entities participated in a requirements gathering session. GE identified needs for demonstration and deployment of the system and the MPR facilitated these activities during the 2009 and 2010 MPRs. RespondComm was able to mount the system on its mobile elevated structure, ensuring operability throughout the MPR.

The RespondComm project was originally funded by NIJ. RespondComm's focus is to evaluate Worldwide Interoperability for Microwave Access (WiMAX) and determine the advantages that such a network provides regarding public safety and emergency response communications. Additionally, the project has expanded to include rapidly deployable tower platforms with integrated alternative power sources to establish critical response networks which leverage the flexibility of the WiMAX equipment.

RespondComm offered back-up power sources, alternative communication mediums, and the ability to mount GE cameras on an elevated, mobile platform.

The Team:

- GE Global Research:
  - Dr. Nils Krahnstoever, Principal Investigator
  - Dr. Ting Yu, Scientist
  - Don Hamilton, Scientist
- Mock Prison Riot
  - Michael Lucey, Project Manager
- RespondComm
  - Bob Chico, Program Manager
  - Dave Ramsburg, Senior Electrical Engineer
  - John Mazzie, Electrical Engineer
  - David Buckingham, Electrical Engineer
- HCS Technologies

  - Bruce Headly, RespondComm subcontractor
  - Dan Headly, RespondComm subcontractor

- Evaluation and Role Player Teams

  - FBOP/FCI Schuylkill (PA)
  - Delaware County (IN) Sheriff's Office
  - Lake Erie (OH) Correctional Institution

- Observers and Subject Matter Experts

  - Mr. Jack Harne, Program Manager, National Institute of Justice
  - Mr. Brian Montgomery, Program Manager, National Institute of Justice
  - Weber State University (UT) - School of Criminal Justice

4.1.3 System Deployment

The MPR facilitated the limited operational assessment of the GE Global Research Intelligent Video System using LEC practitioners during MPR training scenarios and demonstrations from May 3-5, 2010 on the grounds of the decommissioned West Virginia Penitentiary in Moundsville. Preliminary algorithm capture and system testing took place on the grounds on May 1-2, 2010, to determine camera mounting locations and conduct equipment calibration prior to deployment.

The objective of this assessment was to gather general practitioner feedback on the technology in its current form, capture interactions that are typical in a correctional environment and evaluate the current system capabilities on live enactments.

LEC practitioners and role player volunteers were used to obtain the highest degree of "typical" inmate behavior. GE Global Research provided general suggestions on what behaviors the LEC teams could exhibit. LEC practitioners also suggested behavioral activities that would be conducive to the system; hence, several additional, unplanned scenarios were enacted by practitioners and role players. The assessment team provided guidance on how scenarios should be enacted to ensure that practitioners and role players stayed within view of the cameras.

As in 2009, the GE Global Research Team operated two parallel systems during the MPR. One system did not perform any video analysis but rather focused on collecting high-quality, full-resolution, full frame-rate video. This data will be used by the research team to develop and optimize its algorithms. A second system was used to perform data collection while also analyzing videos for events. Because of the computational load of the video analysis, the second system performed video storage at a lower frame-rate.

The North Yard and adjacent basketball court areas of the venue served as an ideal location to execute scenarios. Data was gathered by capturing activities through cameras mounted throughout the area, in addition to camera units mounted on the RespondComm trailer. Compared to 2009 an additional thermal camera was mounted at the site. All scenarios for this evaluation took place in this location under a variety of weather conditions (e.g., sunny; cloudy; at times, heavy rains) and during daylight hours. Scenarios lasted from 3 to 15 minutes.

(a)  (b) 

Figure 4: (a) The Pan-Tilt-Zoom (PTZ) camera mounted in the basketball court area. (b) The GE work station in the North yard area of the MPR campus.

Camera System Calibration: As opposed to MPR 2009, during the MPR 2010 data collection, the calibration of the system was performed with a surveyor theodolite, which greatly improved the ease, speed and accuracy of the calibration process.

4.1.4 List of MPR 2010 Scenarios

1. **Gang Formation:** Two or more "gangs" are occupying the yard area. People mingle in two

separate groups. Only interactions between members of the two groups occur.

Research Question: Can the system detect the presence of two (or more) distinct communities within the prison? Can the system detect the loitering event?

2. **Gang fight:** An argument breaks out between two inmates of opposite gangs. It comes to a confrontation between two opposing factions. A larger fight ensues. Some third party inmates try to stay away from the event. Corrections officers break up the fight.

   Research Question: Can the system detect the initial face-off and raise an alert? Can the system detect the fight? Can the system detect the event of inmates "fleeing" from the action?

3. **Failed Assault:** An inmate is planning to attack another, who sees it coming. He runs and tries to avoid the attacker.

   Goal: Can the system detect the fast moving event or chasing event?

4. **Suspicious event:** Several prisoners are hanging out in the basketball court area. There are two separate groups / gangs. Due to some event (insult, offensive gesture), one gang is deciding to assault the other gang. After some initial planning and deliberation among the members of the aggressive gang, they charge and a fight ensues. Officers intervene and break up the fight.

   Goal: Can the system detect the planning phase of the gang members and raise a warning? Can the system detect the fight?

5. **Assault on officer:** An inmate pretends to get injured during a basketball play. A crowd gathers around the inmate. An officer enters court to investigate and gets ŚjumpedŠ by inmates. It was a trap. The officer calls out for help. Other officers approach to assist. It comes to a stand-off between prisoners (holding the injured officer) and the officers that came for assistance. A fight ensues during which the situation is brought under control.

   Goal: Can the system detect the gathering of inmates around the ‘bait’. Can the system detect the stand-off between the officers and the inmates?

6. **Inmate Stabbing:** Inmates are playing in the basketball court area. Several watchers. One inmate is planning on stabbing or clubbing another inmate. Bystanders see this event coming and proactively move away from the area where the anticipated event is going to occur. The

fight breaks out between the two inmates. People watch from a distance. Officers intervene and break up the fight.

Goal: Can the system detect the "moving-away" event?

7. **Exchange of contraband:** People are playing in the basketball court. In the periphery, one inmate meets with another. They exchange some item (shank, cell-phone, knife).

   Goal: Can the system detect the meeting between two inmates and flag it as an event?

Overall, the following scenarios have been captured as part of the Mock Prison Riot 2010 data collection and testing:

300 Gang Attacking Group II
301 Gang Attacking Group
302 Schuylkill Aggression Between Inmates
303 Schuylkill Aggression Between Two Gangs
304 Two Gangs Hanging Out
305 Two Gangs Walk By
306 Two Gangs Walk By 2
307 Suicide Attempt
308 Contraband (Multiple Types)
309 Slight Agitation
310 Slight Agitation 2
311 One Gang Attacks Another
312 One Gang Attacks Another 2
313 Attack and Chasing
314 Two Gangs Meet
315 Sharpening and Fight
316 Thermal + EO Data Collection
317 Thermal + Smoke Data Collection

4.2 MPR 2010 Example Results

The system was run live in real-time with all algorithms switched on that were developed in response to the data collected at the Mock Prison Riot 2009.

An example result is the processing of sequence "300 Gang Attacking Group". The system managed to correctly detect all event components and the final occurrence of the fight between inmates. The Figure 5 shows the event table the system reported during the processing of the sequence. Figure 6 shows several screen shot from the processed video sequence. Overall, the system managed to successfully detect the presence of multiple groups, predicted the fight and then detected the aggression.

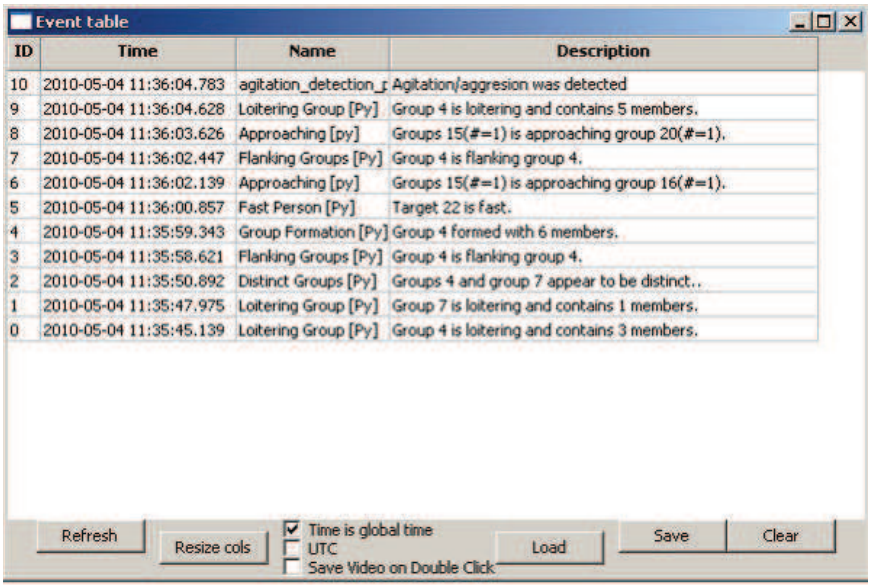


Figure 5: **Example MPR 2010 Aggression Scenario.** This figure shows the event table that the system reported during the processing of the scenario at the MPR event.

4.2.1 Contraband Handoff

One activity between inmates that captures the attention of correctional officers in particular is that of "contraband handoff". Contraband handoff is the exchange of items such as improvised knives, drugs, messages, cell phones and many other similar items. At the 2010 mock prison riot we have again collected data sequences in which correctional teams enacted contraband handoff scenarios

Figure 6: **Aggression Scenario.** Events detected live by the system during an aggression scenario enacted by Lake Erie Corrections. The aggression in the last frame is indicated by the red rectangle that is partially hidden by an overlaying loitering label.

for us. Our approach to handoff detection was previously described in Section 4.2.1. The described algorithm currently has a detection rate of 0.25 at a zero false alarm rate. We will provide ROC curves later in this program.

Figure 7 shows several contraband detection events detected in the Mock Prison Riot data:

4.2.2 Other Scenarios

Based on guidance and requests from correctional teams the following new scenario types were added during the Mock Prison Riot 2010 event

- Knife Sharpening
- Suicide Attempt
- Walk-By

The latter scenario depicts one or several members from one gang purposefully bumping into members of another gang. This is a very subtle event, but of high interest to corrections communities.

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

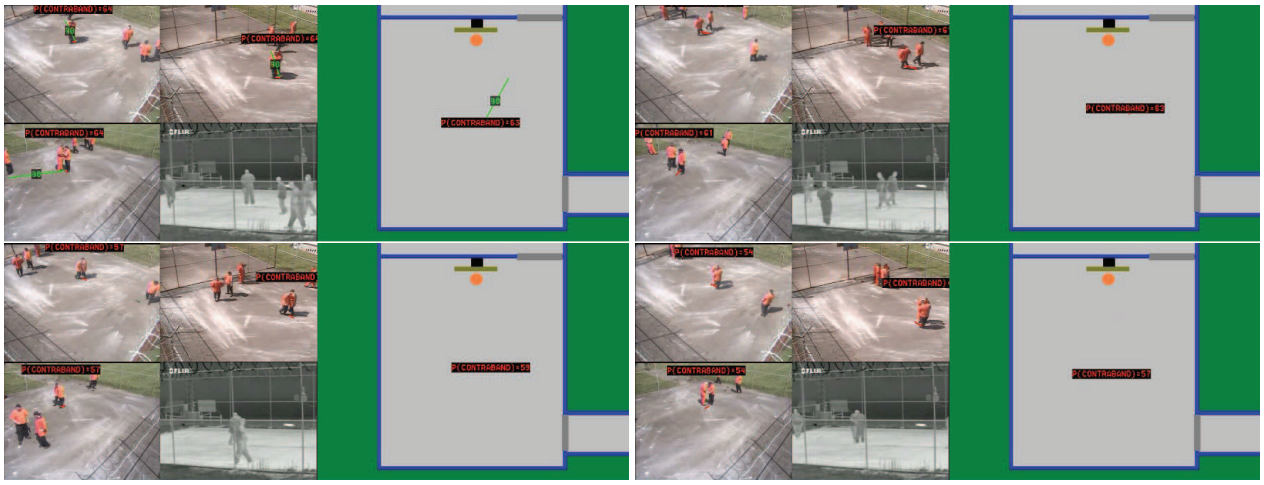


Figure 7: **Contraband Handoff.** The system successfully detected a large number of contraband handoff events in the enacted Mock Prison Riot data.

4.2.3 Smoke

During the 2009 Mock Prison Riot, the use of smoke grenades simulators was found to severely hamper the performance of the tracking system. This was again the case in 2010, however we performed a data collection with a thermal camera to assess the possibility to handle smoky conditions in the future. This encourages the future use of thermal imagers for deployments where smoke might be present.

4.3 Schenectady Police Data

Since June 2011, we have made a number of visits to Schenectady Police Department and interact with the staff managing the Schenectady's Public Surveillance Camera Project (PSCP). During these visits, we have selected a subset of video archives that contain the recording of past events that have already been viewed and considered closed. These video data have been transferred to GRC. We have been using these data as the testing set for our system.

4.4 GE Global Research Collection

We have performed data collection on face capturing from far field at the GE Global Research Courtyard, where four fixed surveillance cameras and four PTZ cameras are equipped.

5 Evidence Representation

In general, automatic surveillance systems will perform behavior recognition and event detection by processing visual observations with algorithms that make decisions about the occurrence of said behaviors or events. Virtually any surveillance system in existence today will perform some kind of event detection, however the problem with existing systems is that they perform poorly in challenging environments (such as crowded prison yards) and that they are difficult to extend. Once a system is deployed, an operator can perhaps control certain parameters for a set of hard-coded algorithm but current system do not allow to easily add new type of behaviors and events to the system. Hence, the goal of the current program is (i) to greatly enhance our ability to automatically recognize behaviors in challenging (i.e., crowded) environments and (ii) to empower operators and users of the system with the ability to create new behavior and event descriptions.

Toward this goal, we need to shift from hand-tailored algorithms that process observations and make decisions about events to a paradigm where general purpose reasoning engines perform cognition on models that encode behaviors in a unified manner. We need to develop methods that allow operators to easily describe such models and abstract the data that such models and reasoning engines can consume . The reason for this is that we will not know during system design time what kind of data or evidence an operator needs for recognizing future behaviors.

Hence, one of the first tasks to be addressed by this program is to unify the representation of all internal system data in a way that it can easily be consumed by future reasoning components and to expose this representation to the system (and operator) in a suitable manner.

We have determined that the problem that needs to be solved here is related to that of the *Semantic Web* [9]. The semantic web is an evolving effort of the internet community to attach machine-readable semantics (i.e., meaning) to all content stored in the world wide web. The problem with the WWW as it exists today is that even though it is easy for a human to read and understand web content, it is exceedingly difficult for a computer to put meaning behind the raw data that is stored on the web. This is the reason why current search engines still focus on providing keyword search capabilities. As an example, if one would like to determine the distance

between two cities in the US, one traditionally focuses on finding a web page that happens to provide distance tables rather than asking a search engine directly for the information that is needed. The reason is that web pages might list pairs of cities and the distances between them, but the raw data does not semantically encode the fact that the page lists pairs of cities, that the cities have geographic locations, that there exists the concept of *distance* between them and that actual distances are provided. The Semantic Web attempts to remedy this problem and defines a set of standards and tools that provide (i) well defined syntax and structure for data (XML Schema), (ii) a method for representing meta data (RDF) and meta data taxonomies (RDFS), (iii) ontologies (OWL), (iv) rules and (v) queries. In particular the quest for meta data representation shares many common concerns with what we need to do to generalize the representation of observations made by our visual surveillance system.

We have hence begun to design a metadata data model that exposes internal data in tree structured registry. With this framework the system can expose a wide variety of data. Examples include the number of individuals currently being visible by the tracking system, the ground velocity of each individual as well as more abstract internal data such as uncertainty information about the location of each individual. Access to this information is no longer restricted to internal C++ code (the programming language that the GE Intelligent Video system is written in). In contrast exposed data can be accessed by any external scripts, simply by specifying the location of the data in the metadata registry. As an example, to retrieve the current groundplane location of the first target, one can retrieve that piece of information with the key

```
grc_intelligent_video::tracking::trackers[0]::state::gp_vx
```

Similarly, if one would like to add information to the existing metadata registry, one can specify

```
grc_intelligent_video::tracking::trackers[0]::prob_agitated = 0.87
```

which in this case could mean that the probability of the individual tracked by tracker 0 to be agitated is 87%. All internal data can be viewed to be represented by a tree structure, where nodes in the tree denote objects and children denote sub-components of these objects, which in itself are values, objects or arrays of objects. Values can be numerical, text, matrices or (for future

reasoning) probabilities and probability density functions. An example snapshot of the metadata tree at a certain point in time during tracking is provided here:

```
grc_intelligent_video
    + type = gesec::multi_cam_process
    + num_views = 4
    + channel
    | +[0]
    | | + frame_nr = 45107
    | | + step_count = 0
    | | + frame_width = 320
    | | + frame_height = 240
    | | ` frame_depth = 3
    | +[1]
    | | + frame_nr = 45107
    | | + step_count = 0
    | | + frame_width = 320
    | | + frame_height = 240
    | | ` frame_depth = 3
    | +[2]
    | | + frame_nr = 45107
    | | + step_count = 0
    | | + frame_width = 320
    | | + frame_height = 240
    | | ` frame_depth = 3
    | `[3]
    |   + frame_nr = 45107
    |   + step_count = 0
    |   + frame_width = 640
    |   + frame_height = 480
    |   ` frame_depth = 3
    + site_state
    |   + next_id = 7
    |   ` targets
    |     +[0]
    |     | + id = 0
    |     | ` history
    |     |     ` trajectory
    |     |       +[0]
    |     |       | + time = 294683621057
    |     |       | ` state
    |     |       |     + gp_x = 1.0205035630178392e+001
    |     |       |     + gp_y = 1.0197667705150886e+001
    |     |       |     + gp_vx = 0
    |     |       |     + gp_vy = 0
    |     |       |     + w = 0.6
    |     |       |     + h = 0.6
    |     |       |     ` l = 1.9
    |     |       +[1]
    |     |       | + time = 294683621064
    |     |       | ` state
    |     |       |     + gp_x = 1.0205035630178392e+001
    |     |       |     + gp_y = 1.0197667705150886e+001
    |     |       |     + gp_vx = 0
    |     |       |     + gp_vy = 0
```

```
|      |        |      + w = 0.6
|      |        |      + h = 0.6
|      |        |      ` l = 1.9
|      |        +[2]
|      |        | + time = 294683621068
|      |        | ` state
|      |        |      + gp_x = 1.0284011291271824e+001
|      |        |      + gp_y = 1.0036370524247703e+001
|      |        |      + gp_vx = 2.4293216567458091e-003
|      |        |      + gp_vy = -4.9615632121974857e-003

... etc
```

Our representation is similar to the subject-property-value representation by RDF [10] except that we don't, at least at this point, require a precise semantic definition of all components or the use of uniform resource identifiers (URI Refs) to link meta data to globally known resource type descriptions. In our framework an example entry is

```
grc_intelligent_video::tracking::trackers[0]::state::gp_vx  = 0.00234,
```

where the subject is `grc_intelligent_video::tracking::trackers[0]::state`, i.e., the object that is being described is the state of tracker 0 in the scope of the tracking sub-system of the GRC intelligent video application. The property is `gp_vx`, i.e., in this case, the x component of the groundplane velocity. Finally, the value is 0.00234. The meta data representation forms the basis for the *Evidence Database* depicted in the schematic overview in Figure 102 as well as for the operator based scenario modelling, a vision of which is depicted in Figure 8.

It is envisioned that the forthcoming behavior recognition framework will consume data from this metadata database and augment the database with additional insights (i.e., data and possibly probabilistic evidence).

5.1 Resource Description

Resource Description Framework: A resource description framework (RDF) [1] that is responsible for dynamically representing and maintaining probabilistic and non-probabilistic meta data about the entire surveillance system has been completed. The RDF creates a registry of name-value pairs that provides a hierarchical representation of system meta-data in memory. The design of the

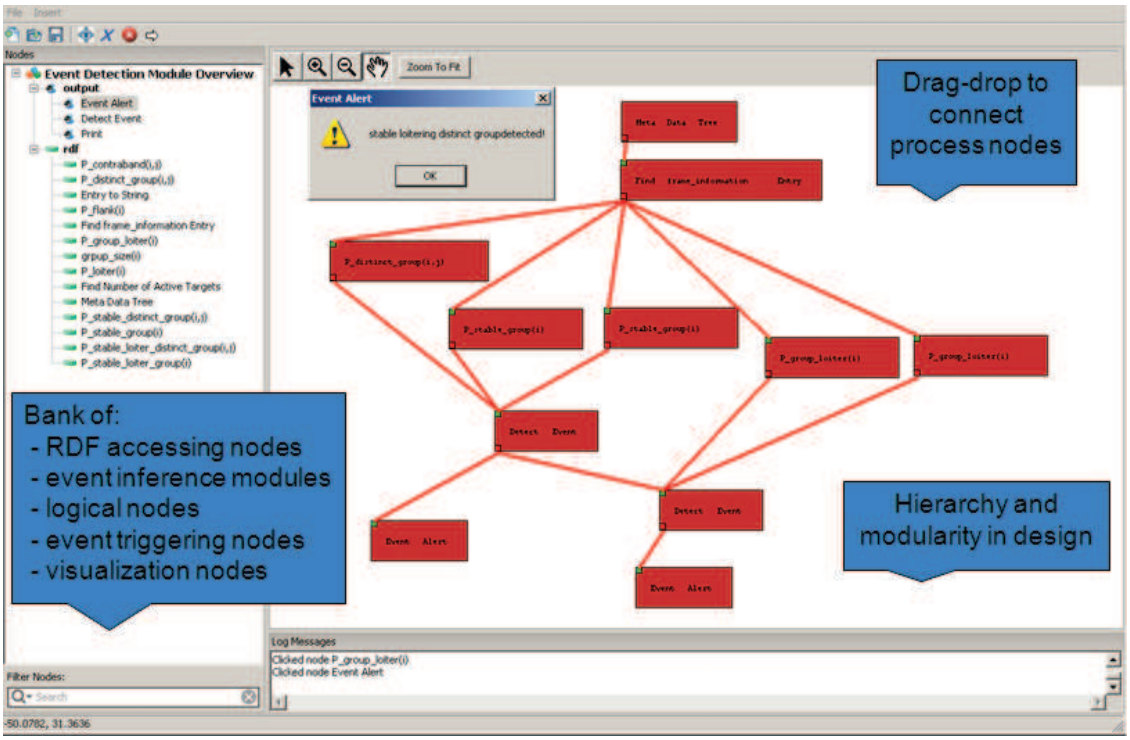


Figure 8: **Scenario Modeler:** The operator will be able to specify scenarios using a convenient modular tool.

RDF is non-declarative in the sense that the consumer of RDF meta data does not have to be aware of the content of the RDF or the type of RDF items. In addition, the RDF is dynamic and can be extended by outside components. These properties of the RDF enable the development of modular "plug-in" components that can query the RDF, perform inference and draw conclusions and then insert meta data that represents new insights into the RDF tree. Details of the RDF design have been provided in the previous quarterly report.

6 Probabilistic Low Level Evidence

A first set of low and mid-level evidence is now obtained from video, following probabilistic approaches. Where previous approaches have relied on heuristics and ad-hoc methods to determine certain properties (e.g., whether an observed target is moving "fast"), the current approach is to rely on sound probabilistic modelling and inference to assess the probability of events (random variables) and probabilistic distribution of continuous random variables. For example, for the detection of "unusually fast" moving targets the system previously made "yes" or "no" decisions based on thresholding the velocity of a target. However, this approach does not reveal any confidence in the decision which is required for letting operators combine uncertain evidence into robust scenario and event recognition components. A principled approach that assesses the probability of a target moving "unusually fast" reveals that a wide range of factors need to be taken into consideration to make this assessment. For example, if there is a priori the possibility that vehicles or spurious fast moving false alarms are observed in the field of view, relying solely on the use of velocity thresholds is inadequate to make an assessment of whether or not an observed target is moving unusually fast. This aspect will be discussed in more detail in the next section.

6.1 Fast Person Detection

The task of simply detecting a fast moving person is a good illustrative example of what it means to utilize a principled probabilistic approach to determining such an event. As indicated in the previous section, the standard approach to simply perform a thresholding operation on the track velocity is not sufficient for several reasons:

- The observed track might follow a target that is not of TargetType = person.
- The observed track might be a false alarm.
- There is no sense of confidence in estimation.

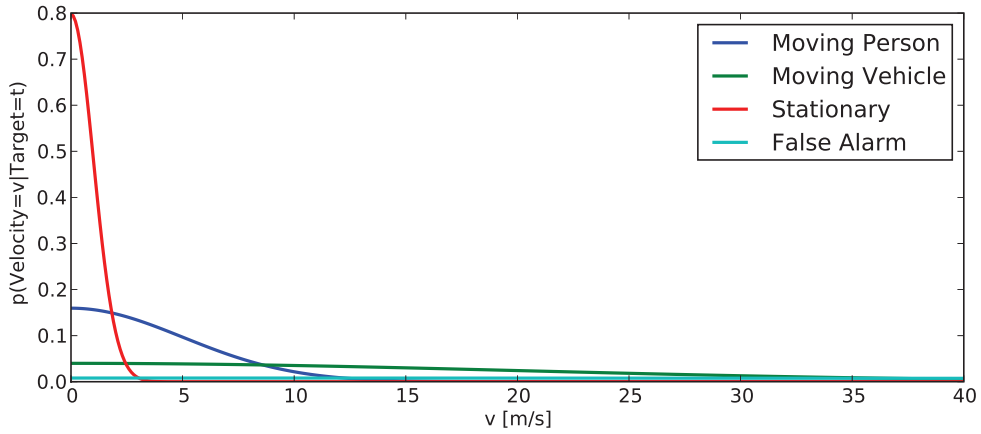


Figure 9: **Velocity Priors:** Prior probabilities of target velocities for different target types.

Hence the assumption is made that a track can be one of four separate categories and provide the prior probability of observing the different track types (Table 1). Hence, for example, given no

TrackType	P(TrackType)
Moving Person	0.6
Moving Vehicle	0.1
Stationary Target	0.1
False Alarm	0.2

Table 1: **Target Type Priors:** Prior probabilities of observing targets of different types.

other information, there is a 10% chance that an observed track is following moving vehicle.

Relationships about the type and expected velocity of a target need to be established. This can be done by estimating (or roughly gauging) expressions for the expected velocity of a target given its type $p(Velocity = v|TargetType = T) = p(v|T)$ (see Figure 9). Furthermore, the notion of what it means for a target to be *fast*, given it's velocity and type needs to be formalized. Here "soft thresholds" that express the uncertainty for a target to move *unusually* fast given it's velocity. The uncertainty not only represents our inability to exactly state when a targets speed is noteworthy. It also "covers up" the presence of noise in the tracking system. For example, a target that is observed to move at a speed v might in reality move slower. An observed high speed can partially be attributed to tracking errors and detection clutter. Figure 10 expresses the models

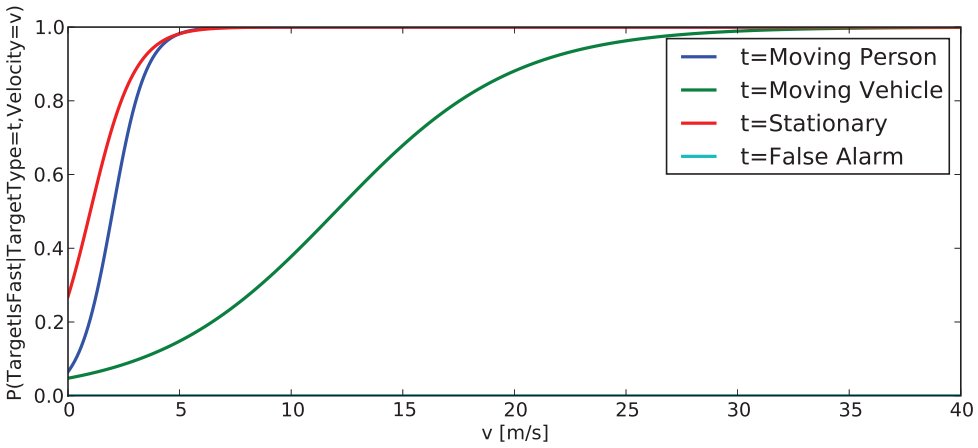


Figure 10: **Soft Velocity Thresholds** Probability of a target being unusually fast, given it's type and velocity. These distributions are modelled with a logistic distribution using the sigmoid function.

for $P(TargetFast = F|TargetType = T, Velocity = v) = P(F|T, v)$, where[1] $TargetFast \in \{True, False\}$. In order to now reach a decision on whether we are observing a person that is moving unusually fast, we need to examining our belief in the fact that we are observing a person (Table 2) and that the observed person is unusually fast (Table 3). These priors, conditional

TargetType	P(IsMovingPerson=True|TargetType)
Stationary Target	0.0
Moving Person	1.0
Moving Vehicle	0.0
False Alarm	0.0

Table 2: **Probability of observing a moving person**.

probability distributions and conditional truth tables are represented by the Bayesian Network in Figure 11. This Bayesian Network can now be used to perform simple queries. For example, given a track for which only it's velocity is known, one can compute probabilistic estimates of what target type is being observed. Figure 12 illustrates this for two different prior probabilities

[1]Here an expression $P(A|B)$ is viewed as a short-hand notation for $P(RndVarA = A|RndVarB = B)$, which in turn can be viewed as shorthand for $P(\{$Event that random variable RndVarA takes on value A$\}|\{$ Event that RndVarB takes on value B$\})$. In cases where a random variable expresses a Boolean value, the expression $P(A|B)$ is usually shorthand for $P(A = True|B) = P(RndVarA = True|B)$.

IsMovingPerson	TargetFast	P(IsFastPerson=True|TargetType,IsMovingPerson)
True	True	1.0
True	False	0.0
False	True	0.0
False	False	0.0

Table 3: **Probability of observing a fast moving person**.

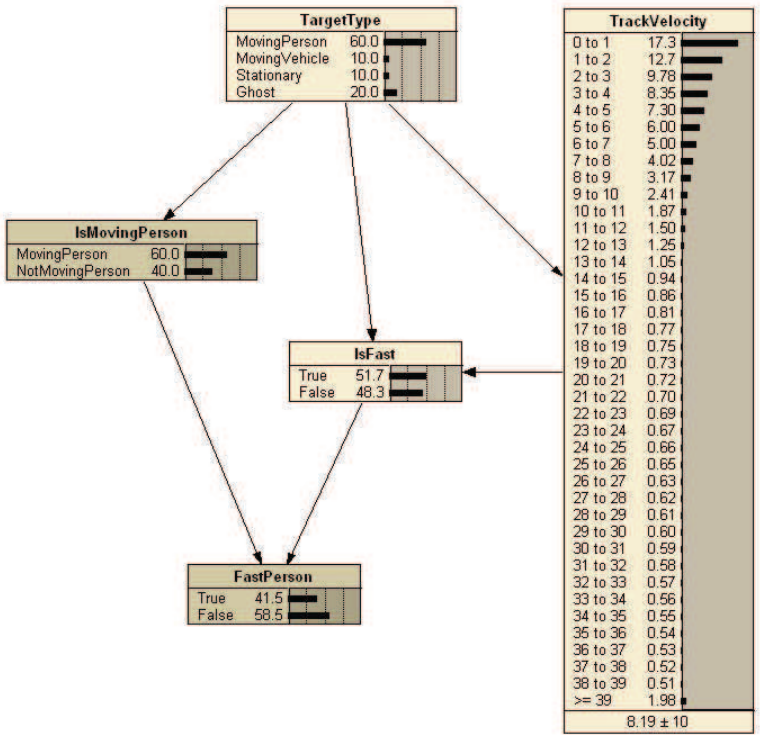


Figure 11: **Graphical Model for Fast Person:** The continuous random variable for velocity has been discretized.

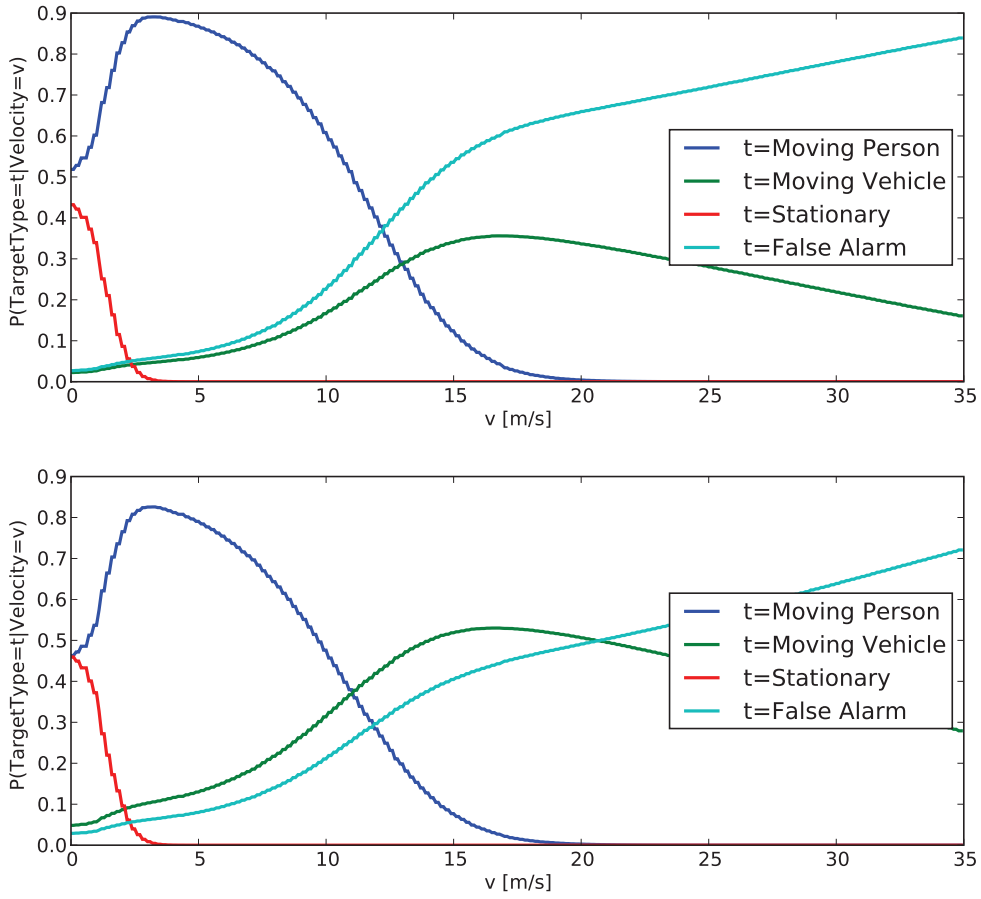


Figure 12: **Estimating the target type by measuring the velocity:** The above graph expresses the probability of seeing the different target types if we only know it's velocity. For the top graph we have assumed a 60% prior probability of observing a person and a 10% prior probability of observing a vehicle. For the bottom graph we assumed a 50% prior probability of observing a person and a 20% prior probability of observing a vehicle.

of observing the different targets. For zero velocity targets the target being a person is the most probable, with the second most probable solution being a stationary target of otherwise undefined type. For higher velocities, the probability of the target being a person is most probable. For even higher velocities the target is most probably a vehicle or a false alarm depending on the a priori probability of seeing vehicles in the first place. If there is almost no chance to observing vehicles at all, the best explanation of observing high velocities is that of a false alarm. People can just not move this quickly. For very high velocities, the false alarm explanation always wins.

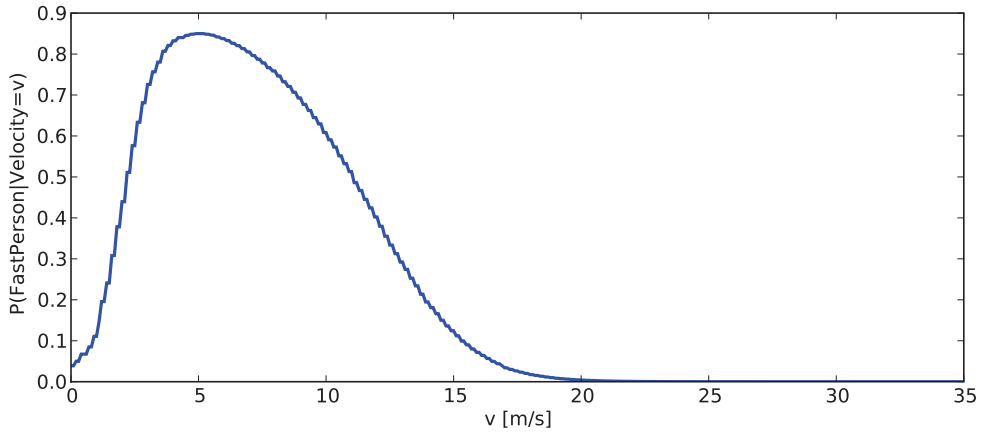


Figure 13: **Fast Person:** Probability of seeing a "Fast Person" given the velocity of the target. We have assumed a 60% prior probability of observing a person and a 10% prior probability of observing a vehicle.

Finally, we can compute the probability of observing a fast person, given an observed target velocity (see Figure 13). Initially the probability is low because the velocity is too small to be considered fast for a person. For higher velocities, the target is most probably a person and also considered fast. For even higher velocities, the target is fast, but most probably a false alarm.

Hence, based on careful modelling of the domain and specification of several simple priors (Table 1), prior probability distributions (Figure 9), and conditional probabilities (Figure 10), we have obtained a capability to perform sound probabilistic reasoning about several important aspects of our system.

6.2 Slow Person Detection

Experiments and empirical observations indicate that the automatic detection of low-level events such as *slowness* have a tendency to fail in the presence of noise or due to violations of the assumptions that are made to compute the evidence. For the case of *slowness*, several factors make a simple assessment based on velocity error prone:

- a target can be slow, but be in the process of acceleration or deceleration
- detection errors (missed detections, poorly positioned detections) especially in the presence

of other targets nearby, can lead to data association errors, which in turn leads to an overestimate of the velocity and an underestimate of *slowness*

Hence we developed *two* measures of slow speed. One is based on looking at the raw velocity estimate maintained by the tracker. The other computes lower-bound velocity over a window of target locations via numerical differentiation. These two velocity estimates are combined with an additional feature that indicates the current *crowdedness* to obtain an estimate of the probability $P(S_{ij} = \{\text{Target } i \text{ is slow at time } t_j\})$. In practice, the accuracy in tracking a target could be affected by the ambiguity of nearby detections. In other words, the success of monitoring a target is affected by the crowdedness of the target under tracking. The concept of crowdedness will be explored in the next section. Figure 14 shows the Baysian Network for modeling and computing this probability. The concept of crowdedness enters the network as evidence *HasTargetsNearby*. Crowdedness increases the belief in the trajectory being noisy, which alters our inference of whether or not we can make a robust assessment over a target moving slowly. A noisy track appears to move faster than a noise-less track. Hence, if a target is **not** considered slow based on the robust velocity estimate and there appears to be no noise in the system, we give a low final confidence to the target being slow (5%). However if the target doesn't appear to be slow, but there is a presence of noise, we give it a higher (20%) probability that it is actually slow. This approach improves the detection of slow moving targets in groups and crowds, where target are often subject to significant association jitter.

As a side effect, the network can infer whether a target is accelerating and whether the target track might currently be subject to noise. The discrete top node called "HasTargetsNearby" is the node representing "Crowdedness".

Figure 15 shows the time-series of slowness in the network in Figure 14 on the sequence that Figure 17 is based on. Observe that during 176s to 185s when the target is slow, it is also determined to be loitering in Figure 17.

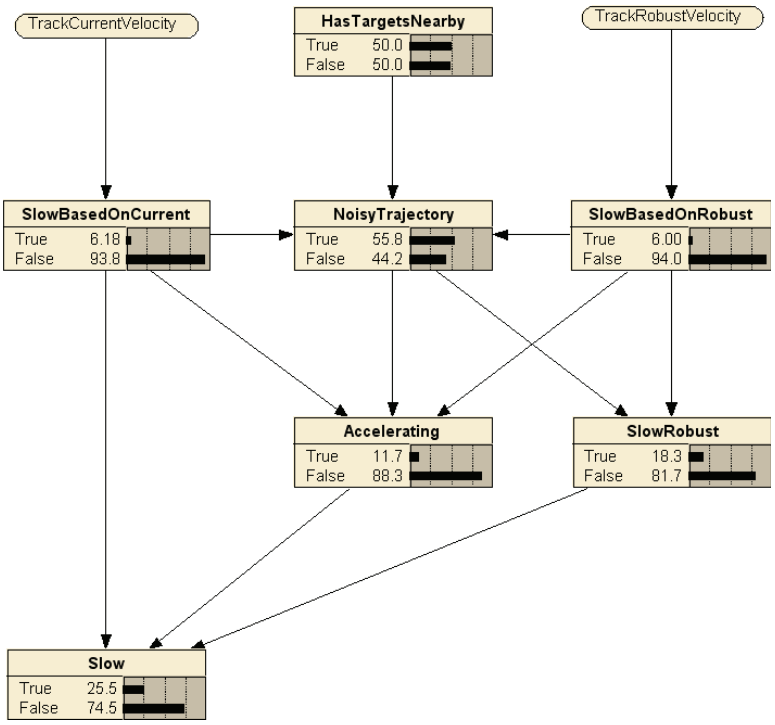


Figure 14: **Probability of being Slow:** A Bayesian network is used to compute the probability $P(S_{ij} = \{\text{Target } i \text{ is Slow at time } t_j\})$.

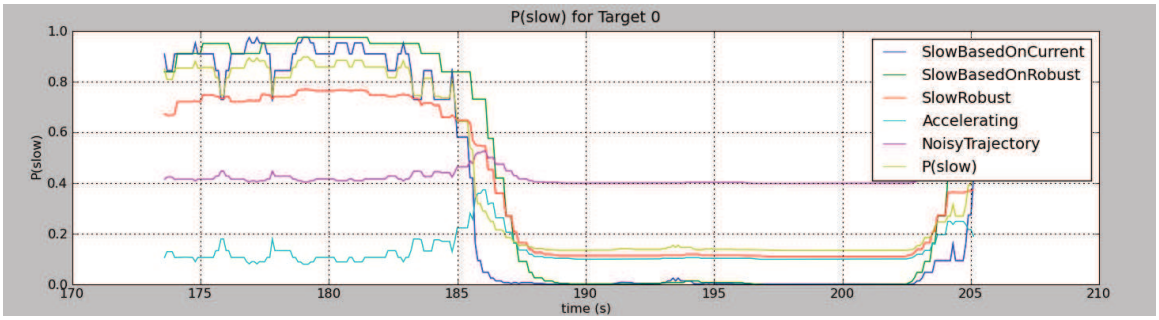


Figure 15: **Slowness Timeseries**. Observe how the slowness corresponds to the loitering timeseries in Figure 17.

6.3 Loitering Detection

An additional low-level event considered is *Loitering* which can be defined as the act of remaining in one place with no apparent purpose. The detection of loitering is important to many municipalities where gang activities and disorderly youth are a problem. For example, Section 1., Chapter 8-4 of the Municipal Code of Chicago defines:

> *"Gang loitering" means remaining in any one place under circumstances that would warrant a reasonable person to believe that the purpose or effect of that behavior is to enable a criminal street gang to establish control over identifiable areas, to intimidate others from entering those areas, or to conceal illegal activities.*

While an intelligent video system is not a good judge of determining the *purpose* of a persons behavior, it nevertheless can aid law enforcement to automatically detect whether a person is remaining in one place for protracted amounts of time. The detection of low-level loitering is the topic of this section.

In order to facilitate a probabilistic analysis of loitering, one needs to examine the variables that influence whether or not a person might be loitering. Here, after some analysis, loitering was defined as an activity that exhibits the following pattern:

- a loitering person moves slowly for extended periods of time (*i.e.*, a person who just stopped is not considered to be loitering)
- a person must be moving slowly to be considered loitering (*i.e.*, a person who is moving fast or running, is no longer loitering)
- the movement pattern is such that the person frequently revisits the same locations (*i.e.*, it is not necessary that the person stays in exactly the same spot, but the person should at most be wandering, frequently revisiting spots the person visited before)

The above description encompasses the concept of the *Slowness* of a person's movement as well as the concept of *revisitations*. A more analytical definition of loitering is the following. A loitering person:

- is currently moving slowly
- has been close to the current position at a point in time in the past that was at least $t_{\min}$ seconds ago and at most $t_{\max}$ seconds ago
- was moving slowly at that previous point in time as well

Hence, for every target T_i at a frame time t_j we examine for all times $t_k \in [t_j - t_{\min}, t_j - t_{\max}]$ in the past whether the person was loitering. We denote a sub event as $L_{ijk} = L_i(t_j, t_k) = \{\text{Target } i \text{ was exhibiting loitering behavior at times } t_j \text{ and } t_k\}$. It is conditioned on a target's trajectory, which we denote as $\mathbf{X}_i = \{(t_j, \mathbf{x}_{ij}) | j = 0, ...\}$. We assess the probability of this event L_{ijk}, namely $P(L_{ijk}|\mathbf{X}_i)$ using the following model:

$$P(L_{ijk}|\mathbf{X}_i) = P(L_{ijk}|S_{ij}, S_{ik}, C_{ijk})P(S_{ij}, S_{ik}, C_{ijk}|\mathbf{X}_i) \tag{1}$$

$$= P(L_{ijk}|S_{ij}, S_{ik}, C_{ijk})P(S_{ij}|\mathbf{X}_i)P(S_{ik}|\mathbf{X}_i)P(C_{ijk}|\mathbf{X}_i), \tag{2}$$

where $S_{ij} = S_i(t_j)$ denotes the event that target i was slow at time t_j and $C_{ijk} = C_i(t_j, t_k)$ denotes the event that the location of target i at time t_j was close to it's location at time t_k. The above relationship is expressed in a simple Bayesian network depicted in Figure 16. Following the verbal description above, we utilize the canonical $\mathrm{NoisyAndDist}()$ conditional probability distribution [11] to define the truth table for $P(L_{ijk}|S_{ij}, S_{ik}, C_{ijk})$

$$P(L_{ijk}|S_{ij}, S_{ik}, C_{ijk}) = \mathrm{NoisyAndDist}(L_i, 0.05, C_{ijk}, 0.2, S_{ij}, 0.2, S_{ik}, 0.2) \tag{3}$$

where $\mathrm{NoisyAndDist}()$ is defined as:

$$\mathrm{NoisyAndDist}(E, p_0, C_1, p_1, \ldots, C_n, p_n) = (1 - p_0) \prod_1^n (C_i + (1 - C_i)(1 - p_i)) \tag{4}$$

The probability $P(L_{ijk}|\mathbf{X}_i)$ is estimated for all $t_k \in [t_j - t_{\min}, t_j - t_{\max}]$. We define the event of

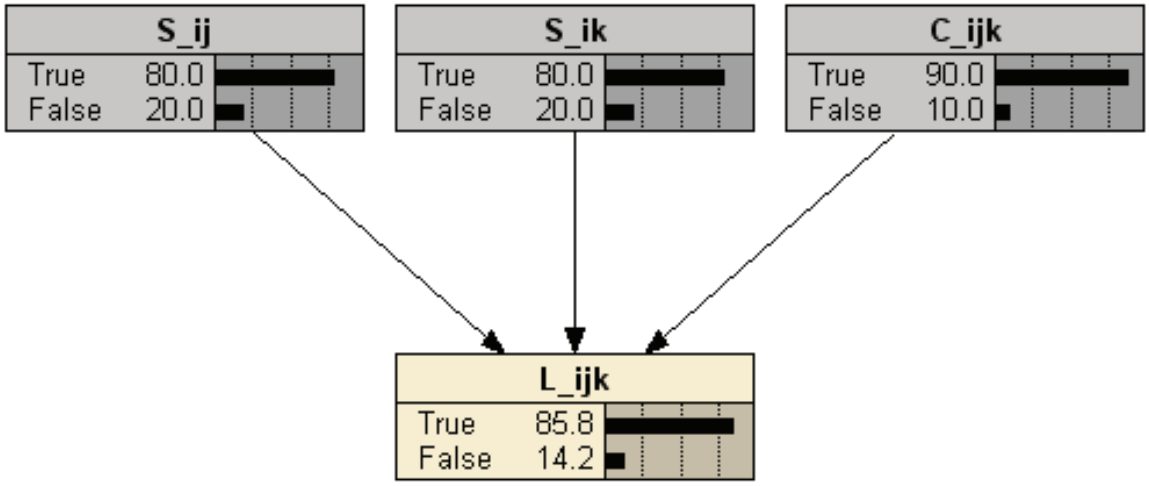


Figure 16: **Loitering Sub-Component**: L_{ijk}.

target i to be loitering at time t_j as $L_{ij} = L_i(t_j)$ and model it's probability as

$$P(L_{ij}|\mathbf{X}_i) = \max_{t_k} P(L_{ijk}|\mathbf{X}_i) = \max_{t_k} P(L_i(t_j, t_k)|\mathbf{X}_i), \tag{5}$$

where the max is taken over all t_k as described above.

To compute this expression, we need to define models for C_{ijk} and S_{ik}. Again C_{ijk} denotes location revisitation, that target T_i at time t_j is close to it's location at time t_k. We model C_{ijk} with a soft-threshold using the *logistic* distribution [12, p.503]:

$$P(x, \mu, \sigma') = \frac{1}{1 + exp(-2\frac{-x+\mu}{\sigma'})}, \tag{6}$$

where x is the variable to be classified, μ and σ' are the expected mean and variance of the inflection point. $P(x, \mu, \sigma')$ can be expressed in terms of the *sigmoid* function $\text{sigmoid}(x, \mu, \sigma) =$

$\frac{1}{1+exp(-\frac{x-\mu}{\sigma})}$ by a scaling of the variance $\sigma' = 2\sigma$ and the fact that $exp(-x) = \frac{1}{exp(x)}$:

$$P(x, \mu, \sigma') = \frac{1}{1 + exp(\frac{x-\mu}{\sigma})} = \frac{1}{1 + 1/\left[exp(\frac{-x+\mu}{\sigma})\right]} \tag{7}$$

$$= \frac{exp(\frac{-x+\mu}{\sigma})}{exp(\frac{-x+\mu}{\sigma}) + 1} = 1 - \frac{1}{1 + exp(\frac{-x+\mu}{\sigma})} \tag{8}$$

$$= 1 - \text{sigmoid}(x, \mu, \sigma). \tag{9}$$

The probability that the locations of target i at time t_j and t_k are close, given the observed trajectory $\mathbf{X}_i$ is then:

$$P(C_{ijk}|\mathbf{X}_i) = 1 - \text{sigmoid}(||\mathbf{x}_{ij} - \mathbf{x}_{ij}||, r_{\text{close}}, \sigma_{\mathbf{x}}), \tag{10}$$

where the parameter r_{close} is the radius that defines the closeness in the context of loitering revisitations and $\sigma_{\mathbf{x}}$ is the variance that represents the noise in the locations $\mathbf{x}$. We address the model for *slowness* S_{ik} in the next section.

Figure 17 shows a test sequence, where loitering was detected for a person at the GRC test site. The person stops for a few seconds in the middle of the yard and then continues to walk away toward the parking lot. The graph shows the temporal evolution for $P(\{\text{Target 0 is Loitering at time } t\})$ across time. As the person starts walking again (around $t = 183s$), the probability drops steadily from $P = 0.73$ until it reaches around $P = 0.18$ during the subjects walk. We will discuss our approach to triggering alerts based on Decision Theoretic methods in later reports.

Figure 18 shows an example from the 2009 Mock Prison Riot where some inmates where detected to be loitering near the fence. In the figure the probability of loitering is denoted for every active track as a green "L" with an opacity that is proportional to $P(Loitering|Evidence)$, *i.e.*, for low probabilities the "L" is transparent.

6.4 Crowdedness Detection

The goal is to reliably reason about the *crowdedness* of an area around a detected target (a person), *i.e.*, making probabilistic decision about whether a target currently has other targets in it's vicinity.

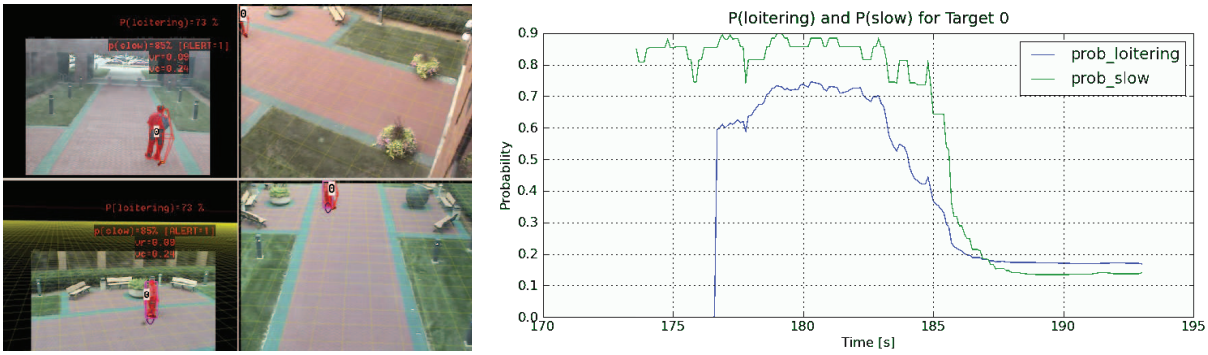


Figure 17: **Loitering Person Example:** Scene showing a loitering person. The scene is shown on the left and the temporal evolution of $P(\{\text{Target 0 is Loitering at time } t\})$ shown on the right.

Figure 18: **Loitering Inmates:** Inmates near the fence of the recreational yard have a high probability of loitering (indicated by green "L"). This figure incidentially also shows the probability of inmates moving slowly (denoted by red "S"), which is the case for all inmates.

The crowdedness is useful for both group analysis in the high-level and for estimating detection confidence in the low-level. Consider that there is detection uncertainty such as false-alarms and missed-detections. For example, it's possible that a person is standing very close to the person in question. If no actual targets are observed, a person is miss-detected. Also, an observed neighbor could be a false target due to detection errors. A track is called *real* if it is not a false-alarm; a track is *alone* if it has no hidden miss-detected neighbors. Probabilities of these parameters are treated as *prior* as follows: [2] Denote with FA_i the event $\{$Track T_i is a false alarm.$\}$ and denote with MD_i the event $\{$Track T_i has a nearby person that is not tracked (*i.e.*, it is a missed track)$\}$. Then it is assumed that

- $P(\mathrm{FA}_i) = const. = 0.1$, *i.e.*, there is 10% chance that a track is false.
- $P(\mathrm{MD}_i) = const. = 0.05$, *i.e.*, there is 5% chance that a track has a missed detected neighbor.

Also, let $P(\mathrm{TP}_i) = P(\neg \mathrm{FA}_i) = P(\{\text{Track } T_i \text{ is a true positive track.}\}) = 1 - P(\mathrm{FA}_i)$. Here *crowdedness* around a track T_i is measured as the probability of a crowdedness event C_i that target T_i has at least one target in it's vicinity. Given observed n neighbors, $P(C_i|\{\text{all other tracks}\})$ is computed as follows. Consider the neighborhood defined by a radius r and a variance $\sigma_{\mathbf{x}}$ that represents the localization uncertainty, the crowdedness of a (real) target $C(T_i)$ can be derived as follows:

$$P(C_i = False|\{\text{ all other tracks}\}) = P(\neg \mathrm{MD}_i) \prod_{j \neq i} P(\{\text{track } T_j \text{ is not "near" track } T_i\}) \quad (11)$$

$$= (1 - P(\mathrm{MD}_i)) \prod_{j \neq i} (1 - P(\{\text{track } T_j \text{ is "near" track } T_i \text{ and } \textit{not} \text{ a false alarm}\})) \quad (12)$$

$$= (1 - P(\mathrm{MD}_i)) \prod_{j \neq i} (1 - P(N_{ij} = True)(1 - P(\mathrm{FA}_j))), \quad (13)$$

[2] The case that the track of interest itself a false alarm is not considered here (which can be modeled separately). Also, the probability of a miss-detection is modeled as a per-unit-area measurement. Here the miss-detection probability is taken to be with respect to the vicinity of each track.

Figure 19: **Crowdedness estimation for the targets under tracking.** The crowdedness value is between 0 and 1, where higher value means more crowded.

where $P(N_{ij} = True)$ is the probability of {Target T_i near Target T_j} and is again modelled by a logic distribution using the *sigmoid* function of the distance d_{ij} and the neighborhood defined by r and $\sigma_{\mathbf{x}}$ (see Eqns.6 - 9) :

$$P(N_{ij}) = 1 - \text{sigmoid}(\mathrm{d_{ij}}, \mathrm{r}, \sigma_{\mathbf{x}}) = 1 - \frac{1}{1 + \exp(-\frac{\mathrm{d_{ij}} - \mathrm{r}}{\sigma_{\mathbf{x}}})}. \tag{14}$$

Figure 19 shows an example result of crowdedness estimation using Eqns.11 and 14.

6.5 Group Formation and Dispersion

Detecting and modeling group-related events are important for many high-level behavior analysis for law enforcement including group formation, meeting, dispersion, and following, *etc.*. Given a set of targets under tracking, the goal here is to compute a probability estimate of the likelihood when a group is formed. In order to robustly estimate the variety of ways a group can form, one need to evaluate the probabilities of all relevant group sizes, and such estimation should depend on all individual targets that make up a group. Consider a simple scenario consisting of 3 targets T_i, T_j, and T_k, there are 5 possible configurations: *(i)* no group forms, *(ii)* T_i and T_j form a

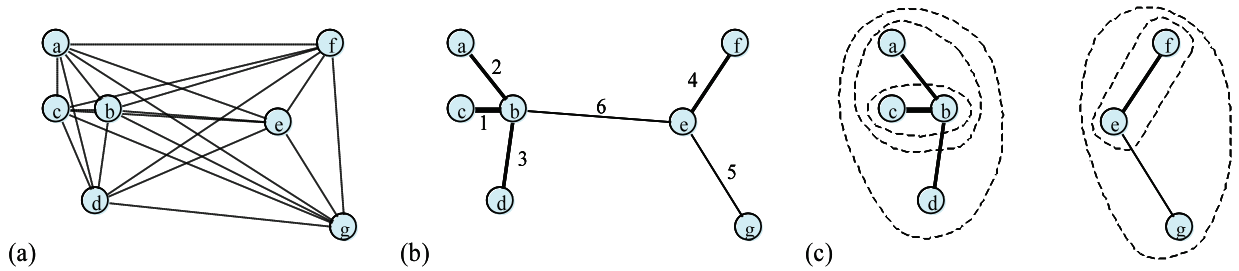


Figure 20: **Determine group formation following a minimum spanning tree (MST) construct.** (a) The combinational nature in determining the grouping of several targets (a to g). (b) We group together nearby targets using the MST of the targets. Edge weight reflects the distance between targets. (c) A "hierarchy" of groups is created following exactly the steps the MST is constructed using Kruskal's algorithm by grouping together disjoint forests (see text).

group, *(iii)* T_j and T_k form a group, *(vi)* T_j and T_k form a group, *(v)* all 3 targets form a group. Figure 71 illustrates another example of 7 targets in determining their grouping. As one shall see, the combinational complexity in grouping grows exponentially as more and more targets in the vicinity are considered. Also, the grouping event should be monitored dynamically in the video and thus it is required to be evaluated in a per-frame basis. To ease the burden in computation, we propose to adopt a clustering approach. Specifically, we group together targets following the *minimum spanning tree* (MST) [13, Ch.24] of the targets. To avoid combinational explosion, we create cluster of sub-groups to form super-groups and construct a "hierarchy" of groups. Our approach is capable of determining all proper grouping configurations by adding nearest targets one-by-one following a consistent rule. Our approach is redundancy free. The computation for group estimation only needs to perform once for all queries, and the algorithm can be carried out efficiently.

We motivate our strategy in grouping targets as follows and show how this leads to a group clustering that is exactly the hierarchical grouping suggested by the MST. We start with definitions and describe the properties of the grouping. A group g is called m*-group of threshold* θ_m if it contains m targets in it, where all pairwise distances between all targets in g do not exceed θ_m. One important property of a group defined by such distance threshold is that it should be *transitive*: if targets T_i, T_j, T_k is a 3-group of θ_k, any pair of targets in it is a 2-group of θ_k. Note that the

decomposition of a group does not follow this rule in the reverse. Instead it follows the triangular inequality of the distance metric: if targets T_i, T_j is a 2-group of θ_{ij} and T_j, T_k is a 2-group of θ_{jk}, then T_i, T_j, T_k should be a 3-group of at least $\theta_{ij} + \theta_{jk}$.

MST group clustering. As the targets are clustered into groups by first considering the closest pair, then the second closest (between next targets or intermediate groups), and so on, the grouping follows precisely the Kruskal's algorithm [13, Ch 24.2] in constructing the MST of the targets. As Figure 71 illustrates, the intermediate groups and the hierarchy of subgroups corresponds exactly the disjoint forest sets in performing Kruskal's algorithm.

Probabilistic formulation. The probability if two targets T_i and T_j is a 2-group is estimated by comparing the distance using the sigmoid function similar to Eqn.9.

$$P(G_{ij}) = p(\{\text{Targets } T_i \text{ and } T_j \text{ form a 2-group}\}) \tag{15}$$

$$= P(\mathrm{TP}_i)P(\mathrm{TP}_j)P(N_{ij}) \tag{16}$$

$$= P(\mathrm{TP}_i)P(\mathrm{TP}_j)\left[1 - \mathrm{sigmoid}(\mathrm{d_{ij}}, \mathrm{r}, \sigma_{\mathbf{x}})\right]. \tag{17}$$

For a 3-group of T_i, T_j, T_k, WLOG we assume $d_{ik} > d_{jk} > d_{ij}$, $p(G_{ijk})$ is decomposed into the probability of a subpgroup $p(G_{ij})$ and the probability whether G_{ij} and T_k is a group:

$$P(G_{ijk}) = P(G_{ij})P(\mathrm{TP}_k)\left[1 - \mathrm{sigmoid}(\mathrm{d_{jk}}, \mathrm{r}, \sigma_{\mathbf{x}})\right]. \tag{18}$$

Similarly for a 4-group T_i, T_j, T_k, T_l, and WLOG $d_{il} > d_{jl} > d_{kl}$ and $d_{ik} > d_{jk} > d_{ij}$:

$$P(G_{ijkl}) = P(G_{ijk})P(\mathrm{TP}_l)\left[1 - \mathrm{sigmoid}(\mathrm{d_{kl}}, \mathrm{r}, \sigma_{\mathbf{x}})\right]. \tag{19}$$

The formulation extends for a group of any size m in general, following a MST construct:

$$P(m\text{-group}) = P((m-1)\text{-group}) \times P(\mathrm{TP}_m) \times \left[1 - \mathrm{sigmoid}(\mathrm{d_{m,m-1}}, \mathrm{r}, \sigma_{\mathbf{x}})\right], \tag{20}$$

where $d(m, m-1)$ is the farthest distance between group $m-1$ and the last target T_m to join the group. Finding $d(n, n-1)$ for the n targets in $\{T_1, ..., T_m\}(1 \leq n \leq m)$ corresponds to constructing the MST following Kruskal's algorithm, which works by first sorting all pairs of distance between the m targets and construct the minimum spanning (disjoint) forest of the closest targets, up to a distance threshold θ_m.

We illustrate the executing of the MST group clustering algorithm using the example in Figure 71, where the grouping of the targets (a to g) is to be determined with a threshold θ_d = 5.5. In our experiments we use $r = 2.5$ and $\sigma_{\mathbf{x}} = 0.5$. The MST algorithm first clusters together targets T_a, T_b, T_c, T_d following edges $\overline{bc}$, $\overline{ab}$, $\overline{bd}$ into 3 groups G_{bc}, G_{abc}, G_{abcd}. It then clusters together targets T_e, T_f, T_g following edges $\overline{ef}$ and $\overline{eg}$ into 2 groups G_{ef}, G_{efg}. Edge $\overline{be}$ has length greater than θ_d and thus no group shall form for the two clusters, then the algorithm stops. The MST is computed efficiently in $O(E \ln V)$, where V is the total targets and $E = V(V-1)/2$ is the number of edges of a complete graph of V nodes. The MST can be efficiently represented using an adjacent matrix M, where the edges of the complete graph is the upper diagonal matrix of M.

So far the MST group assignment answers queries such as the probability a target T_i being part of group of size m. Since the construction of a MST is necessary, finding the group assignment of any target is computationally equivalent to finding the assignment of *all* targets. The next issue is, one target can be assigned to many groups. To answer query like "What is the group size of target T_i?", we go through all assigned groups of T_i (via the MST) and compute the average probability and average group size. For example in Figure 71(c), target T_b has three associated groups and target T_e has two groups. One can fine-tune the average measurements by adopting a histogram analysis and find the 'peak' value of the group probability and the group size out of the soft assignments. Figure 21(a) illustrates the group probability estimation over the Mock Prison Riot 2009 dataset. Figure 21(b) shows the group size estimation over a video collected at the GRC test site.

On the triggering mechanism of the group formation event, we want to avoid frequent triggering of events during the formation of a large group, where inevitably there must be several sub-events

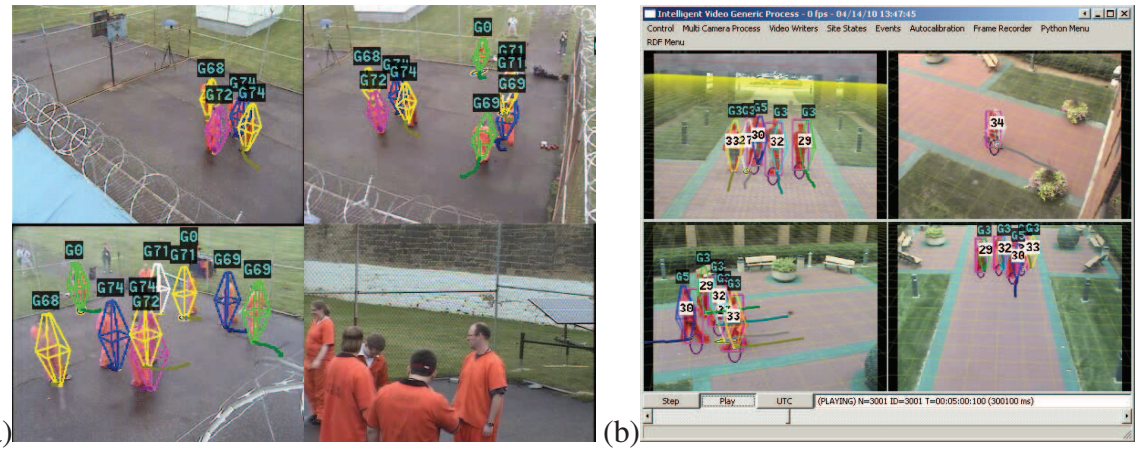


Figure 21: **Probabilistic group detection and group size estimation.** (a) Tracked targets with group formation probabilities. (b) Tracked targets with estimated group size, where transparency indicates the likelihood of the groups.

when forming smaller groups. By further considering the "time" a group sustains, we can rule out false cases such as one target passing by another (*e.g.* and not to trigger a group formation event and immediately canceling it). We are also working at queries like "Are any two specified people in a group?" and to maintain a tracking history of groups along the temporal axis. Such information is valuable in performing further reasoning and explanation of high-level group-related events such as meeting and dispersion.

6.6 Track Healthiness

An important factor influencing the low and mid level tasks that we have described so far is the **reliability** of the tracks. While the estimation of these tasks have been designed to deal robustly with noise in the tracks through soft approaches, the reliability of the tracks remains critical to the success of these algorithms. To this end, our goal is to design an algorithm that is capable of providing an estimate of the health of a given track, even in the presence of occlusions. Conceivably, since our tracking system utilizes a Kalman filter to update the tracks, one could have drawn from the corresponding state covariance an estimate of the health of a track and be done with it.

However, if we consider that a given track is made up of individual detections that have been associated with it, then it become apparent that the reliability of each detection plays a critical

role in the health of the track. Additionally, an estimate of the continuity of *"transitioning"* from one detection to the next determines the consistency of the track's dynamics. Statistically, given a track T and a set of detections $d_1, d_2, ..., d_\delta$ associated with T, we can capture this notion by marginalizing over the detections:

$$P(T) = \sum_{f=1}^{\delta} P(T|d_f) * P(d_f), \tag{21}$$

where $P(T)$ denotes that track T is a healthy track, and $P(d_f)$ denotes that d_f is a good detection. Here, we consider the last δ detections assigned to the track, with the latest denoted as d_δ.

Eqn. 21 sets the stage for capturing several desirable factors that influence the health of a given track. The marginalization encapsulates the probability of each detection $P(d_f)$, while also allowing us to model the consistency of the track given the set of detections $P(T|d_f)$. In specifics, we model $P(T|d_f)$ as

$$P(T|d_f) = P_{link}(d_f|d_{f-1}), \tag{22}$$

assuming that the transition from d_{f-1} to d_f, denoted as P_{link}, is a Markov process. Under such a model, younger tracks would also tend to have a lower degree of healthiness due to the summation aspect of the marginalization. Yet, if a young track has a strong set of detections, the model is still flexible enough to assign it a good healthiness score.

To formulate P_{link}, we specifically consider the frame gap between two consecutive detections and how the consistency of the track changes as we move from one detection to the next. For the former, we wish to model the fact that the confidence we have in a track should decrease proportionally if a pair of consecutive detections are far apart temporally. For the latter, we are penalizing new detection assigned to the track if it causes a drop in the consistency of the track. Let the probability that the track is consistent given a new detection be $P_{cons}(d_f)$. Let the probability that a track is healthy given the frame gap between the newly added detection and the previous detection be $P_{gap}(d_f)$. We can then express them as

$$P_{cons}(d_f) = \prod_{d=x}^{y} 1 - \text{sigmoid}\left(\frac{cov(f,d) - cov(f-1,d)}{\sigma_v}\right),$$
$$P_{gap}(d_f) = 1 - \text{sigmoid}\left(\frac{time(d_f) - time(d_{f-1})}{\sigma_g}\right), \quad (23)$$

where $cov(.,.)$ is a function that provides the variance of the d (x or y) component of the ground plane velocity of the track when a detection was captured, and $time(.)$ gives the time at which it was captured. σ_v and σ_t are the variances "allowed". At the time of this report, we are contemplating a more principled approach for $P_{cons}(d_f)$ based on eigen-analysis of the covariance matrix. Future reports will comment on the status of this development.

The product of $P_{cons}(d_f)$ and $P_{gap}(d_f)$ provides us with a measure of the confidence of the new detection that we incorporate into $P(T|d_f)$ as follow

$$P(T|d_f) = \frac{P_{cons}(d_f)P_{gap}(d_f)}{\sum_{\tau=1}^{\delta} P_{cons}(d_\tau)P_{gap}(d_\tau)}. \quad (24)$$

6.6.1 Detection Probability

The probability of each individual detection, $P(d_f)$ in Eqn. 21, plays an important role in the health of the track. In complex scene, occlusion often leads to false detections, and more detrimentally, false detections that are subsequently assigned to a track. Additionally, we are concerned about false alarms arising from sudden lighting change and shadows. One should avoid initializing a track based on a false detection as well as assigning a false detection during tracking.

Normalized cross correlation. In order to overcome such difficulties, we have incorporated into the meta infrastructure a probability score based on the Normalized Cross Correlation (NCC) [14] that quantifies how likely a detection is not a result of sudden lighting variations in the background. In a nutshell, we utilize the NCC to determine whether a detection is similar to the background, a common characteristic of detections caused by sudden lighting changes and shadows. We have found this to be effective for this purpose.

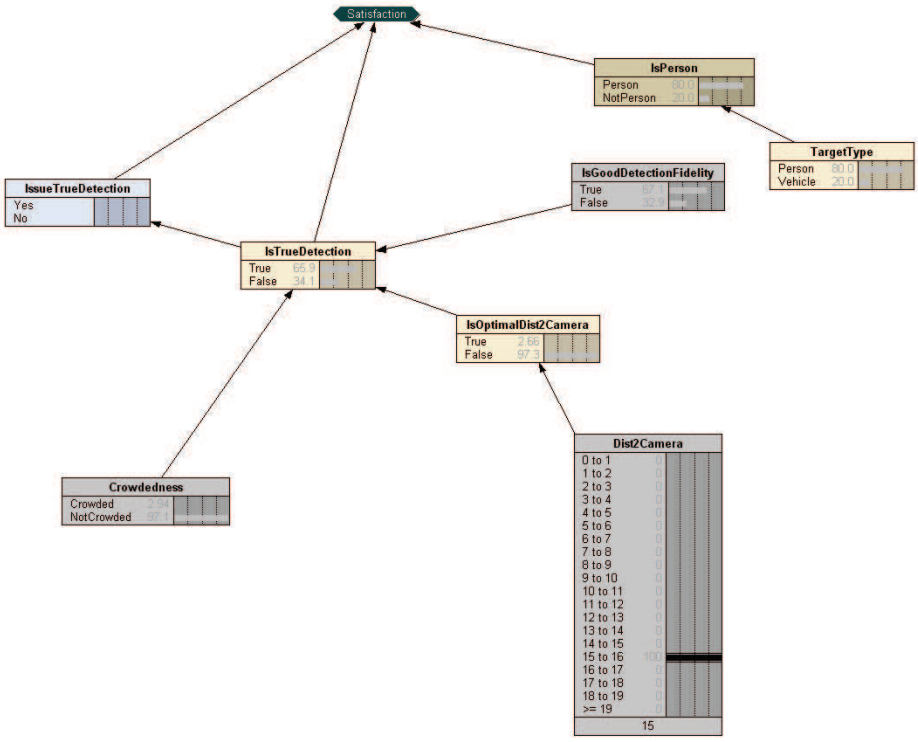


Figure 22: **The Bayesian Network for computing the detection probability.** We model the distance to camera criterion as a continuous node and use it to influence the discrete node "IsOptimalDist2Camera". The other two criteria, "IsGoodDetectionFidelity" and "Crowdedness", represent the NCC score and crowdedness surrounding the detection respectively. These three nodes are combined in the "IsTrueDetection" node, in which combinations of their possible states are used to construct the conditional truth table. Here, we also model the fact that we are only interested in a reliable person detection.

Crowdedness. To deal with false detections caused by crowded situations, we estimate the crowdedness surrounding a detection by examining its spatial relationships to other detections in the vicinity.

Distance from camera. Finally, as the distance of the detection from the camera view increases, we expect a lower detection probability. For a far object, its 3D location is sensitive to slight changes in its image position, and since our tracking system works in 3D, this is a concern for us and has to be modeled.

We combine all the three aforementioned factors in a Bayesian Network shown in Figure 22 in order to make a robust estimate of the detection probability.

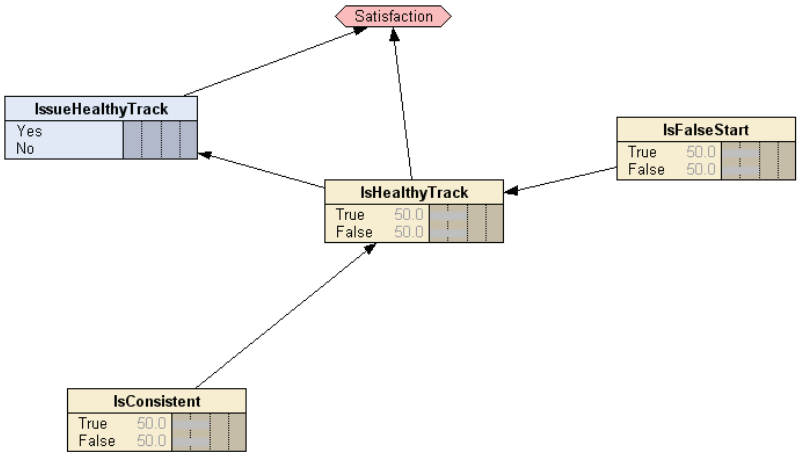


Figure 23: **Healthy Track Bayesian Network.** The node "IsFalseStart" here refers to the probability that the first detection that initiated the track might be a false detection. "IsConsistent" node refers to the consistency of the track computed by Eqn. 21. All possible outcomes of the "IsConsistent" and "IsFalseStart" nodes are modeled in the "IsHealthyTrack" node.

6.6.2 Healthy Track Bayesian Model

The computed value from Eqn. 21 is directly an indication of the health of a track. Additionally, we take into account the possibility that the track might have been initiated with a detection that has a low probability. A track initiated by a false alarm that manage to survive should be given a low healthiness even if subsequent detections assigned to it might have higher probability in some instances. To achieve this purpose, we proceed to combine the track consistency provided by Eqn. 21 and the probability of its first detection in a Bayesian Network shown in Figure 23.

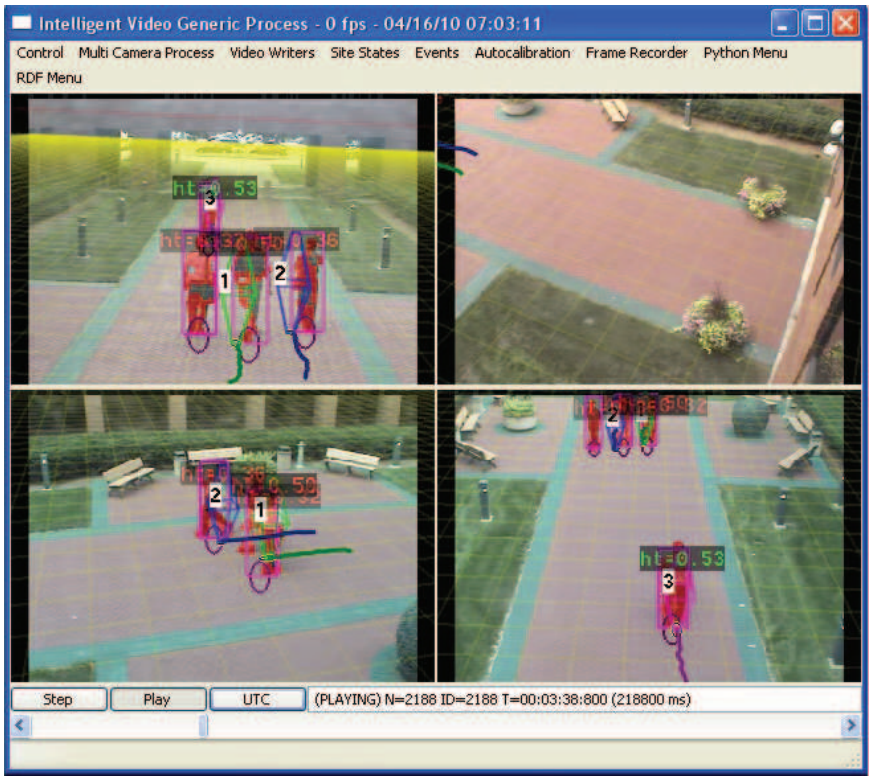


Figure 24: **Example of Track Health Estimation.** In this example extracted from a video taken at the GRC test site, we show how the three tracks close to one another has low track health (indicated by the red boxes above their heads) because they have started out close to one another and remain so up to this point. In contrast, target 3 was started with a strong detection, and had very low crowdedness and good detections throughout. The system recognized that and assigned a higher health score in green.

7 Probabilistic Group Analysis

Defining a precise *grouping* of a crowd is challenging due to the complex social interactions and relations that are hard to measure. To handle the uncertainty in video tracking, we avoid explicit group segmentation and instead maintain a probabilistic measure.

7.1 Pairwise Grouping Measure

We first seek an instantaneous pairwise group affinity measure (that represents the probability of a pair of people belonging to a group), by checking if two individuals are physically close. Inspired by standard social norms from Hall's *proxemics* theory [2] for modeling inter-person spacial relations and the social force model [3] for modeling pedestrian dynamics, we define a pairwise grouping measure based on three main terms: the *distance* between two individuals, the *motion* (body pose and velocity) and the *track history*, as illustrated in Fig.91. How this direct connection probability can be extended to express group membership of non-direct neighbors will be described in §7.2.

The above pairwise grouping measure is defined straight from track observations, thus it favors people that are spatially close. Consider that affinity between people is not always isotropic, our individual-centric affinity is not so, either. Denote d_{ij} the Euclidean distance between two people i and j located at $\mathbf{x}_i$ and $\mathbf{x}_j$ on the ground plane (Throughout the paper, symbols i, j, k refer to a

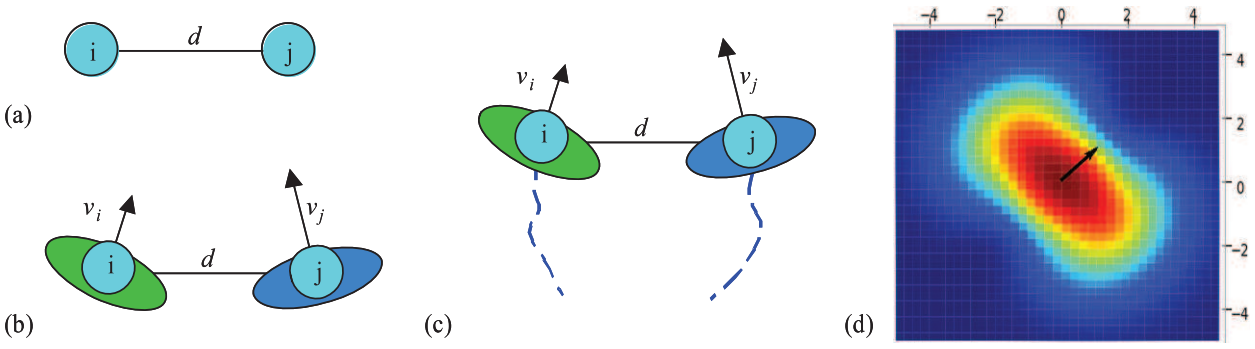


Figure 25: Pairwise group affinity measure: (a) Inter-person distance. (b) Distance and motion (velocity magnitude, direction, frontness/sidedness). (c) Distance, motion, and track history. (d) The instantaneous affinity measure of an individual at $(0, 0)$ with velocity vector $(1, 1)$ in an arrow. Color map depicts this probability kernel between 0 (blue) and 1 (red).

track of a person). We assume a person always faces one's motion direction and denote with ϕ_{ij} the angle between i's velocity vector and the relative position vector $\mathbf{p}_{ij} = \mathbf{x}_j - \mathbf{x}_i$ *w.r.t.* person j. In addition, the affinity varies with the velocity magnitudes of i and j. Overall the instantaneous affinity measure for some time t is hence a function

$$p_c^{\text{inst}}(i,j) = f(d_{ij}, \phi_{ij}, ||v_i||, ||v_j||), \tag{25}$$

where subscript c stands for connectivity, and the dependency of time t is made implicit for clarity. Fig.91(d) visualizes our concrete measure hidden behind the abstract definition (Eq.86). Notice the probability is higher on the side of a person than in the front or back, which is a direct implementation of the aforementioned social norm that states people in a group are more likely to walk side-by-side.

To incorporate track history for robust estimation, we take account p_c^{inst} at time t over a window of T seconds (*e.g.* $T = 3$):

$$p_c^p(i,j;t) = \omega_1 \;\; p_c^{inst}(t) + \omega_2 \;\; \frac{\sum_{t_i \in T} p_c^{inst}(t_i)}{|t_i \in T|}, \tag{26}$$

where ω_1, ω_2 adjust the weights between the two terms of current status and the entire window history ($\omega_1 + \omega_2 = 1$). This improves overall robustness and avoid treating a sudden "passing by" event as an abrupt group change.

7.2 Path-based Group Connectivity

The pairwise affinity measure $p_c^p(i,j;t)$ is defined for two individuals i, j, independent of all other people in the crowd. Observe that two arbitrary individuals in a group do not necessarily have to be directly connected. Rather, it is sufficient that a connecting chain of bonds exists. Here we introduce a path-based group connectivity that estimates the pairwise grouping probability under the influence of others. We say that i and j are *connected*, if there exist pairwise connected intermediate individuals $i_0, ..., i_N$:

$$p_c^{\pi}(\{i \text{ and } j \text{ are connected via } i_0, ..., i_N\}) = p_c^p(i, i_0) \left[\prod_{k=0}^{N-1} p_c^p(i_k, i_{k+1})\right] p_c^p(i_N, j). \tag{27}$$

We then set the connection probability between i and j to be the optimal path amongst all possible paths, which yields the highest probability:

$$p_c^{\pi}(\{i \text{ and } j \text{ are in same group}\}) = \max_{\text{all paths } P_k} p_c^{\pi}(\{i \text{ and } j \text{ are connected via path } P_k\}). \tag{28}$$

To find the optimal path, we first define the edge weight of the initial connection graph to be $\mathbf{G_0}(i, j) = -\log(p_c^p(i, j))$, whose values ranges from 0 to ∞. We then use Floyd's algorithm [15, Ch.32] to compute the all-pair shortest path in $O(n^3)$, where n is the number of tracked individuals. The resulting graph $\mathbf{G_0^{\pi}}$ contains non-negative path weights. We then obtain the final probabilistic connection graph by $\mathbf{G} = p_c^{\pi}(i, j) = \exp(-G_0^{\pi}(i, j))$, where $p_c^{\pi}(i, j)$ is the path-based grouping probability.

The intuition behind this is that the grouping of (i, j) should directly depend on the path created by other individual k in between them. Our path-based metric could be viewed as a simplified solution of a more sophisticated flux-based model, where the connectivity between all pairs of individuals is formulated as a flow, and consider the accumulated flux as the grouping connectivity using the standard *maximum-flow, minimum-cut* algorithm [13, Ch.27]. However the computational cost for the flux-based metric is high (exponential). Our algorithm is also inspired by the spectral clustering [16] and path-based clustering [17] in the domain of pattern classification [18, Ch.10.9].

In case an explicit grouping is desired, we can adopt a proper graph cutting method on $\mathbf{G}$ such as using the hierarchical agglomerative clustering (*e.g. minimum spanning tree* (MST) [19]) or *modularity-cut* [20]. Fig.93 visualize $\mathbf{G}$ (in transparent edges) as well as some explicit grouping (in color polygons) in our test scenarios. Group segmentation is more robust if the hard decision

is made only at the last stage of grouping. Our path-based grouping is less bias than the MST-grouping, since all pairs of paths are considered, whereas in [19] weaker connectivity are ignored in the clustering process.

We will show in the next section that we can perform many reasoning tasks using **G** without explicit grouping: counting the number of individuals in a group, determining if a group is forming or dispersing, modeling the movement of a group, and at a high level if separate groups are about to engage in aggressive activities such as a fight — all using similar probabilistic reasoning steps.

Detailed and formal exposition of the probabilistic group analysis is in Appendix E.

8 Scenario Recognition

8.1 Pairwise Track Relationship Analysis

In addition to the insight about the group structure and group relationships between individuals, it is necessary to reason about the movements and movement patterns of individuals. Hence we have adapted previously developed concepts to the new scenario recognition approach developed in this work. In the following, we will describe the low and mid-level estimation of three key aspects:

- the walking types of people (standing, walking, running)
- the relative pair-wise direction of people's travel (same direction, opposing, neither opposing nor same)
- pairwise changes in distance (decreasing, increasing, same) between people

The above concepts are important components for analyzing events such as *chasing*, *meeting*, *following* and many others in a modular fashion.

8.1.1 Motion Type Detection

We consider three motion types $MT = \{STANDING, WALKING, RUNNING, UNKNOWN\}$. After specifying the priors for the motion type $p(MT)$ and the velocity, given the motion type $p(v|MT)$ we obtain a posterior distribution $P(MT|v)$ shown in Figure 26.

8.1.2 Motion Direction Detection

Given the low level motion of two targets in the ground plane, we analyze the velocity vectors to obtain higher-level probabilistic assessments of their relative motion. More specifically, we allow the high-level relative direction types $DT = \{SAME, OPPOSITE, NEITHER\}$. The direction types are conditioned on the angle between the two velocity vectors between two people.

The graphical visualization of the posterior is shown in Figure 27.

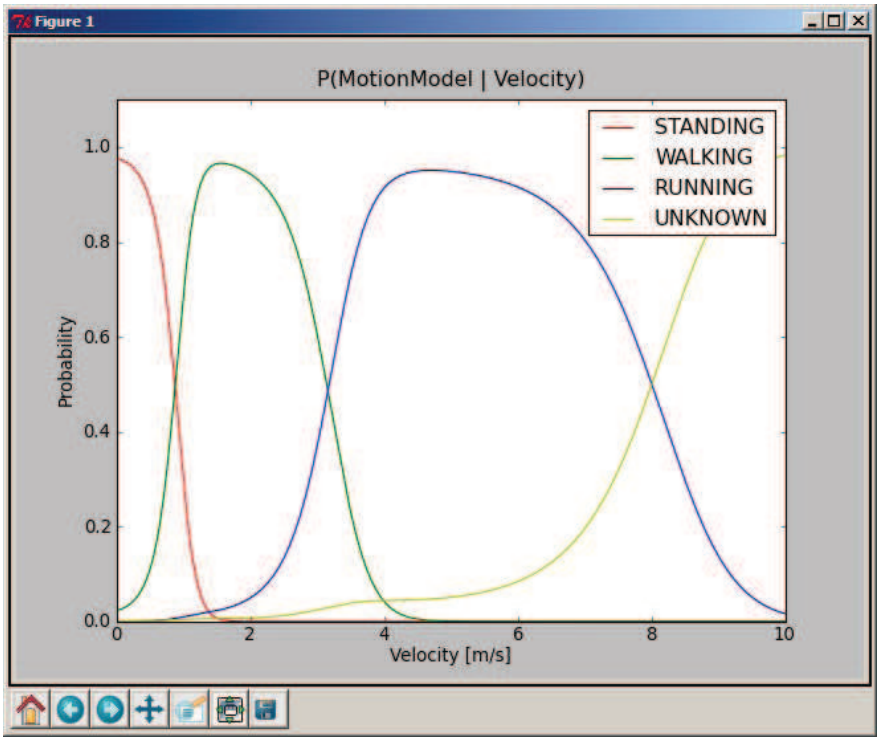


Figure 26: **The probability of a person being in one of three motion types, given the ground-plane velocity.**

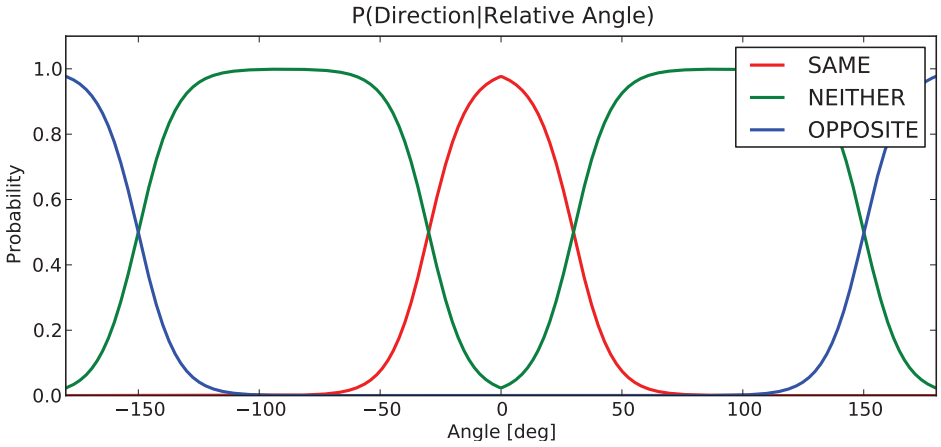


Figure 27: **The probability of two people moving in different ways relative to each other.**

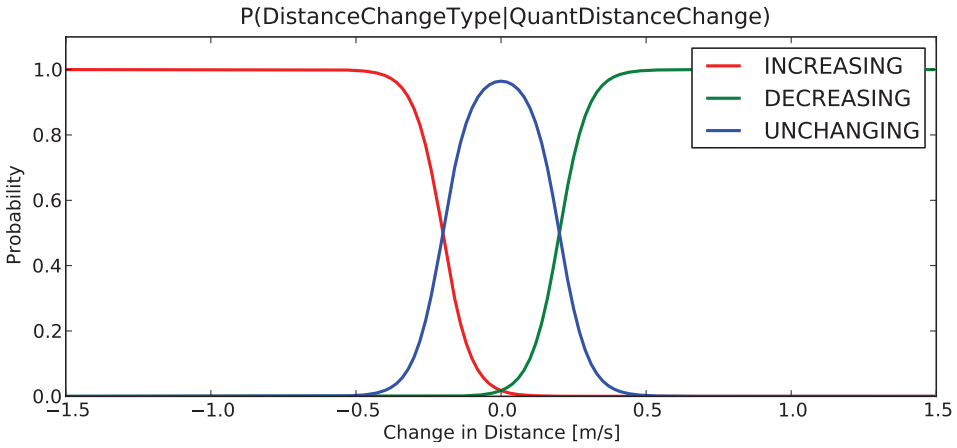


Figure 28: **The probability of different distance change states between two people, given the change in Euclidean distance.**

8.1.3 Distance Change Detection

Similar to the above, we model the change in relative location between two people with the high-level variables $DC = \{INCREASING, DECREASING, UNCHANGING\}$. The assessment of the state given the change in Euclidean distance between two people is shown in Figure 28.

8.1.4 Location Prediction and Convergence

The above three concepts only describe at a coarse level, how people move and how they move relative to each other. Another important concept that is more difficult to assess is the question of whether two people, that move relative to each other are predicted to converge at the same location at some point in the future. The problem here is that a good prediction needs to take uncertainty about the current motion of people into consideration. If it is unclear how two people are moving (e.g., with regards to current location and velocity), it is more difficult to say whether two people are about to converge in the immediate future than if their instantaneous motion is known precisely.

We solve this convergence problem in two ways. First, we utilize the linear state estimation filter that drives the tracker for correctly predicting the state and state uncertainty forward in time. The result is a time varying prediction of location probability distributions, one for each active

target maintained by the system, Second, we will investigate pairs of such distribution predictions on what the probability is that two targets that follow these distributions will spatio-temporally be close.

In terms of the first step, the standard state update equations of our tracking filter are

$$\mathbf{x}_{t+1} = \mathbf{F}_t\mathbf{x}_t + \mathbf{w}_t \text{ with } \mathbf{w}_t \sim N(0, \mathbf{Q}_t) \tag{29}$$

$$\mathbf{z}_t = \mathbf{H}_t\mathbf{x}_t + \mathbf{v}_t \text{ with } \mathbf{v}_t \sim N(0, \mathbf{R}_t). \tag{30}$$

where $\mathbf{x}$ is the state vector (in our case $\mathbf{x} = [x, y, v_x, v_y]$, $\mathbf{z}$ is the projection of the state vector into measurement space (in our case the ground plane), $\mathbf{F}$ is the linear matrix that propagates the state from one time step to the next, $\mathbf{Q}$ is the system noise, $\mathbf{H}$ is the measurement matrix which projects the state vector onto the observation space and finally $\mathbf{R}$ is the measurement noise. In this Kalman filter, the system dynamics $\mathbf{Q}$ is what introduces uncertainty into the system (which in the Kalman filter is reduced by the introduction of measurements).

If we express the state covariance in our Kalman filter as $\mathbf{P}$, the state and the state covariance are predicted as follows

$$\mathbf{x}_t = \mathbf{F}_t\mathbf{x} \tag{31}$$

$$\mathbf{P}_t = \mathbf{F}_t\mathbf{P}\mathbf{F}_t^T + \mathbf{Q}_t. \tag{32}$$

These quantities can be projected into the measurement space as follows

$$\mathbf{z}_t = \mathbf{H}\mathbf{x}_t \tag{33}$$

$$\mathbf{S}_t = \mathbf{H}\mathbf{P}_t\mathbf{H}^T, \tag{34}$$

where $\mathbf{H}$ is the (time independent) measurement model.

The system noise in the above equation is in our system set as:

$$\mathbf{Q}_t = \begin{pmatrix} \mathbf{Q}_t^r & 0 \\ 0 & \mathbf{Q}_t^r \end{pmatrix}, \text{ with } \mathbf{Q}_t^r = q \begin{pmatrix} \frac{t^3}{3} & \frac{t^2}{2} \\ \frac{t^2}{2} & t \end{pmatrix}, \text{ and} \tag{35}$$

$$q = 2\sigma_m^2 \tau_m. \tag{36}$$

To summarize, we can predict the current state $\mathbf{x}$ and state uncertainty $\mathbf{P}$ into the future to obtain at time t the estimated ground plane location $\mathbf{z}_t$ and ground plane location uncertainty $\mathbf{S}_t$. Figure 29 shows an example of these predictions for different motions.

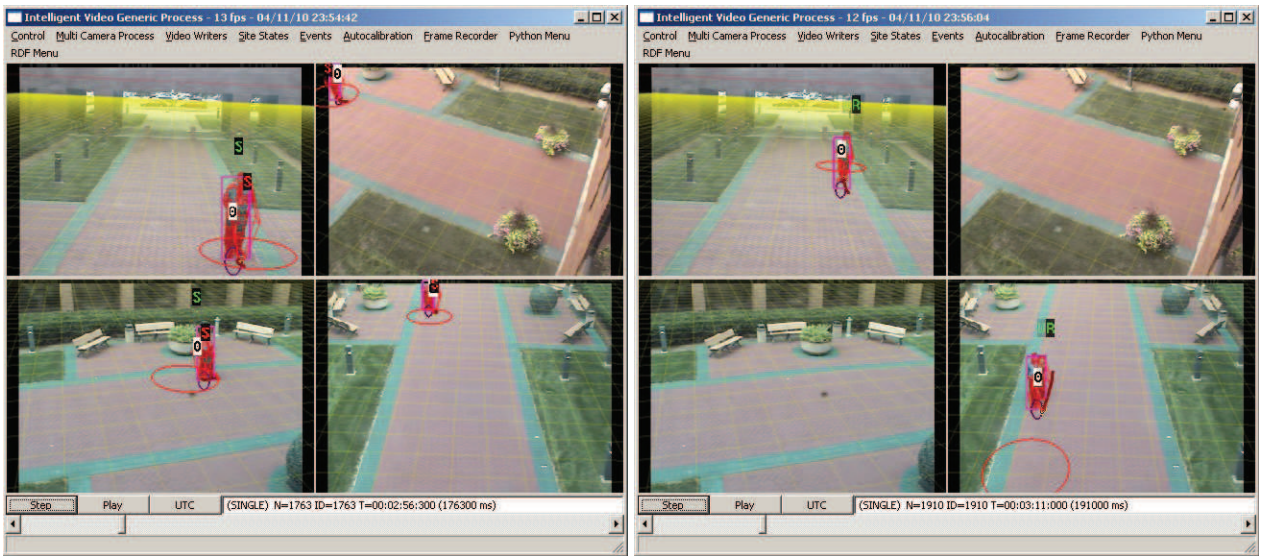


Figure 29: **Prediction of future location.** The figures show the predicted location (indicated by red ellipse) $t = 2sec$ from now for the case of a person standing (top) and running (bottom).

Given the ability to predict future locations, we can now investigate whether two people can be predicted to meet. We assume that the future location of target A and B are given as $\mathbf{x}^A(t) \sim N(\mathbf{z}_t^A, \mathbf{S}_t^A)$ and $\mathbf{x}^B(t) \sim N(\mathbf{z}_t^B, \mathbf{S}_t^B)$ respectively. We then ask what is the probability that the two targets A and B are "close" enough to each other to be considered a meeting event. If we assume that the true locations of A and B are $\mathbf{x}^A(t)$ and $\mathbf{x}^B(t)$ the answer is deterministic and given as

$$p(\{A \text{ and } B \text{ are close at time } t\}|\mathbf{x}^A(t), \mathbf{x}^B(t)) = \theta(||\mathbf{x}^A(t) - \mathbf{x}^B(t)|| - \sigma_c). \quad (37)$$

However, we do not know the exact location of A and B at time t. We hence need to integrate over all possible locations:

$$p(\{A \text{ and } B \text{ are close at time } t\}|\mathbf{z}_t^A, \mathbf{S}_t^A, \mathbf{z}_t^B, \mathbf{S}_t^B) = \quad (38)$$

$$\int_{\mathbf{x}^A}\int_{\mathbf{x}^B} p(\{A \text{ and } B \text{ are close at time } t\}|\mathbf{x}^A, \mathbf{x}^B) \quad (39)$$

$$p(\mathbf{x}^A|\mathbf{z}_t^A, \mathbf{S}_t^A)p(\mathbf{x}^B|\mathbf{z}_t^B, \mathbf{S}_t^B)d\mathbf{x}^A d\mathbf{x}^B = \quad (40)$$

$$\int_{\mathbf{x}^A}\int_{\mathbf{x}^B} \theta(||\mathbf{x}^A - \mathbf{x}^B|| - \sigma_c)N(\mathbf{x}^A; \mathbf{z}_t^A, \mathbf{S}_t^A)N(\mathbf{x}^B; \mathbf{z}_t^A, \mathbf{S}_t^A)d\mathbf{x}^A d\mathbf{x}^B \quad (41)$$

which can be approximated with a set of sample points and weights that represent the distributions $N(\mathbf{x}^A; \mathbf{z}_t^A, \mathbf{S}_t^A)$ and $N(\mathbf{x}^B; \mathbf{z}_t^B, \mathbf{S}_t^B)$ (i.e., via numerical integration) as

$$p(\{A \text{ and } B \text{ are close at time } t\}|\mathbf{z}_t^A, \mathbf{S}_t^A, \mathbf{z}_t^B, \mathbf{S}_t^B) = \sum_i \sum_j (\theta(||\mathbf{x}_i^A - \mathbf{x}_i^B|| - \sigma_c)w_i^A w_i^B). \quad (42)$$

We hence have an expression for the meeting probability between A and B at time t. Removing the time dependency to determine the probability of meeting in (say) a future time interval $t \in [0, T]$ is **not** easily accomplished by marginalizing over the time t. Locations $\mathbf{x}^A(t)$ and $\mathbf{x}^B(t)$ are not independently distributed between nearby time steps t and $t + dt$. We hence chose to select discrete time slices t_i and ask the question of how probable it is that two targets meet at least at one time t_i for a set of times $\{t_i|i = 0, \ldots, N-1\}$. This is the same as the 1 minus the probability that the targets *never* meet and hence given as:

$$p(\{A \text{ and } B \text{ are going to be close}\}|\mathbf{x}^A, \mathbf{P}^A, \mathbf{x}^B, \mathbf{P}^B) = \tag{43}$$

$$1 - \prod_i (1 - p(\{A \text{ and } B \text{ are close at time } t_i\}|\mathbf{z}_{t_i}^A, \mathbf{S}_{t_i}^A, \mathbf{z}_{t_i}^B, \mathbf{S}_{t_i}^B)). \tag{44}$$

8.2 Moludar Inference

We can now infer new events by drawing on the above low-level evidence modules.

Evidence Storage: To recap earlier discussions, all evidence that is estimated in probabilistic and non-probabilistic fashion is stored in a flexible data structure that is accessible from within our visual surveillance system as well as from outside components, in particular dynamically executed scripts. Figure 30 shows a conceptual visualization of this storage process.

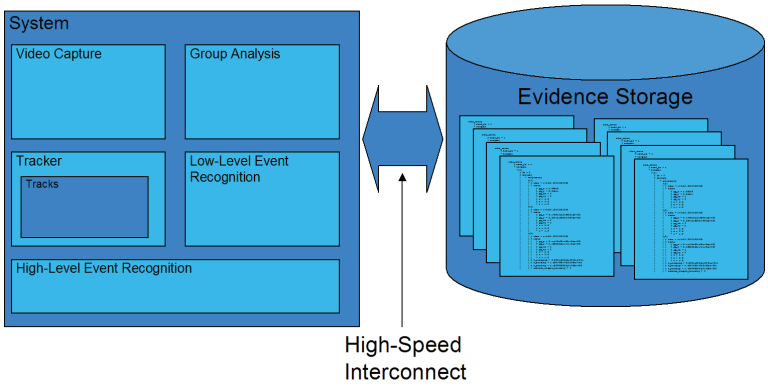


Figure 30: **Evidence Storage** The high-performance tracking system continuously stores evidence in a centralized database.

Reasoning Modules: The modularity in our system is enabled through small modular scripts or C++ based algorithm components that can pull information out of the database and put new inferred information back (see Figure 31). Modules can be aggregated into more complex modules and again be used as part of other components.

Inference Example: In the previous section we presented a collection of powerful probabilistic evidence for several low-level cues: the directionality between pairs of tracks, their motion types, the likelihood of tracks to be meeting in the future and their change in relative location. In the

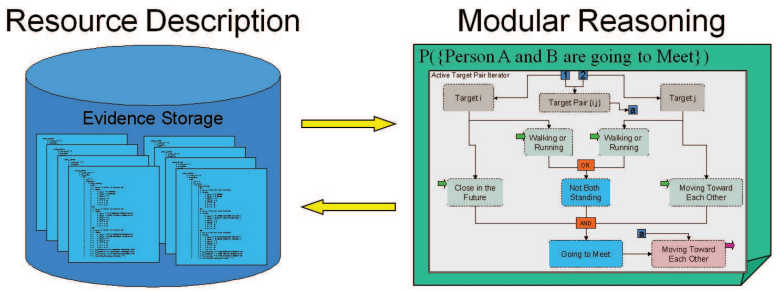


Figure 31: **Modular Reasoning** A high-level view at the concept of modular reasoning. Modules represented as scripts pull information out of the evidence storage, combine it to infer new information and store this new information back into the database.

following we will denote the above evidences as $P_{\text{dir}}(D|A,B)$, $P_{\text{motion}}(M|A)$, $P_{\text{dist}}(G|A,B)$, and $P_{\text{meet}}(C|A,B)$.

The probability of two targets quickly running toward each other to met can now be expressed as:

$$P(\{\text{A and B running toward same location}\}|A,B) = \tag{45}$$

$$[P_{\text{dir}}(D = \text{OPPOSITE}|A,B) + P_{\text{dir}}(D = \text{NEITHER}|A,B)] \cdot \tag{46}$$

$$P_{\text{motion}}(M = \text{RUNNING}|A) P_{\text{motion}}(M = \text{RUNNING}|B) \tag{47}$$

$$P_{\text{dist}}(G = \text{DECREASING}|A,B) P_{\text{meet}}(C = \text{True}|A,B). \tag{48}$$

Here we assume that the conditional A and B represent information known about targets A and B that is needed to compute the given distributions.

The above low-level probabilistic evidence is stored in the evidence storage on a per frame basis. For each frame (i.e., time-slice) of execution a section is added to the "frame_information" sub-array in the metadata tree (see Figure 32).

This representation allows to access the current values of the evidence as well as past values from previous frames and time steps.

Modular Representation: The evidence from the previous section is used in scenario models to

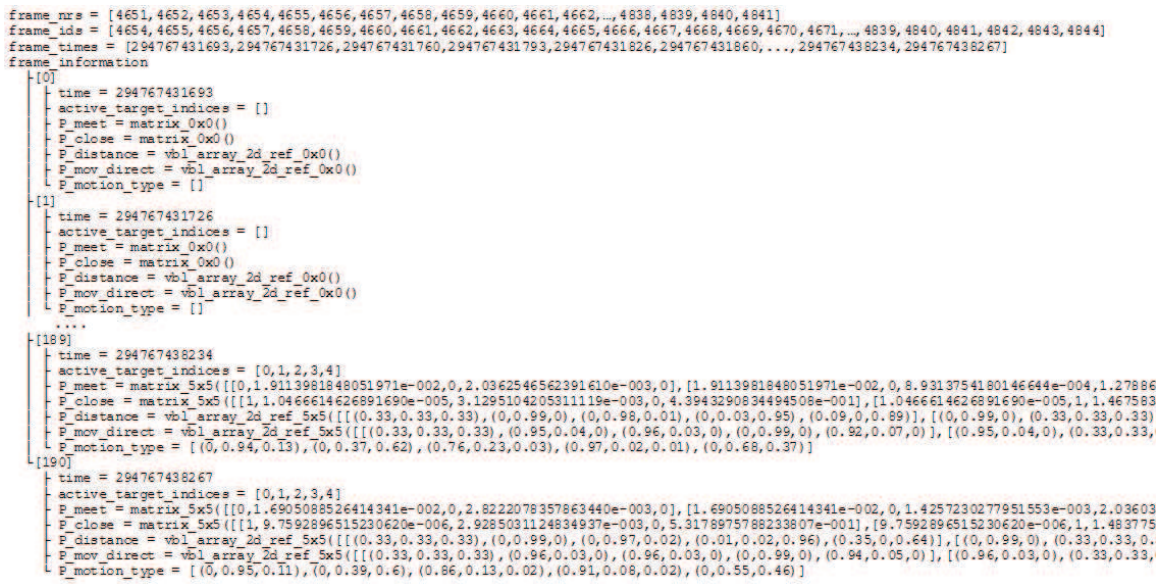


Figure 32: **Representation in Meta Data Tree.** The data presented in the previous sections is stored in a section "frame_information" in the meta data tree.

detect and recognize elusive behaviors in video.

In the current prototype scenarios modeled and defined using a scripting language (Python) that allows to retrieve information from the metadata tree, infer new information and save the new insights back to the tree.

A visual representation of such a scenario can be seen in Figure 33. The shown model graphically represents the "Meeting between two People" scenario, that utilizes the low-level modules described in the previous section.

It was decided to pursue the use of Bayesian Networks (or more generally Graphical Models) as the method for performing inference. Due to the nature of the data that needs to be processed in surveillance, it is necessary to handle both discrete as well as continuous random variables (evidence). Example of evidence includes:

- The velocity $||\mathbf{v}||$ of a target is a continuous non-negative random variable.
- The type of a target (e.g., $\mathbf{TargetType} = \{person, vehicle, false\text{-}alarm\}$) is an example of a *discrete* random variable that assumes one of three states.

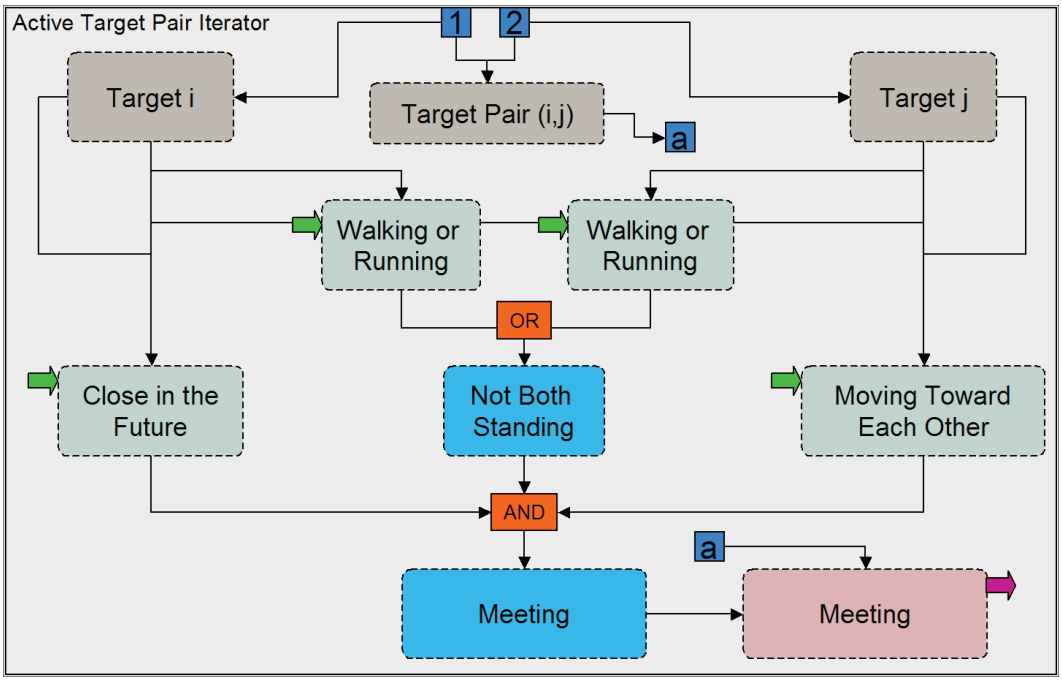


Figure 33: **Prototype Scenario Modeler** The figure represents a prototype scenario modeler. The structure of the scenario model is currently encoded in scrip form using a dynamically typed scripting language.

Hence it is necessary to pursue the use of hybrid Bayesian networks [21, 8, 22, 23] or discrete Bayesian networks where continuous variables are suitably discretized [24].

At this stage we have implemented an initial version of a Bayesian network inference engine using the *junction-tree* algorithm, which is a widely used method to perform exact inference on Bayesian network with discrete node states [12, Ch.14]. The initial version allows the creation of nodes that represent evidence, performs graph moralization, triangulation, and junction tree construction. Belief propagation and inference is performed on the junction tree. As a comparative approach and to allow for convenient visualization of networks in this work, a commercial Bayesian Network system called Netica is also used in this work.

Dynamic Event Scripting: In order to allow our current intelligent vidoe system to be extended with new behavior recognition capabilities, an infrastructure for dynamically loading new behavior scripts was implemented. More specifically, our system was augmented with the ability to create "plug-in" components, written in the programming language Python [25]. Python is a sophisticated

dynamically typed object oriented programming language that trades efficiency of execution for far superior expressive power. Using Python scripting capabilities, developers are able to create new algorithms and software components with greater ease and greater speed compared to a paradigm where system development takes place in C++. In addition, Python system components can be loaded during run-time, alleviating the need to restart a system when new system components are added.

The ability to execute behavior recognition algorithms in Python is a first step towards the ability to have operators modify and change the underlying collection of behavior recognition modules that are executed by the system.

As a first step, we have translated a core collection of low level event and behavior recognition components into Python that have previously been developed under the NIJ program "Detection and Prevention of Criminal and Disorderly Activities" [26].

- agitation detection
- fast person detection
- group flanking detection
- group following detection
- group formation detection
- group loitering detection
- stable group detection
- multiple stable loitering group detection
- stable loitering group detection

Figure 34 shows a screen-shot of the configuration screen that is generated by the Python based "fast person detector".

8.3 Group-Level Scenario Recognition

We describe the key concept toward the flexibility and robustness of our approach, that is to *represent and reason group-level activity on an individual basis using the soft grouping graph* **G**. This

Figure 34: **Dynamic Event Detector Scripts:** Event detection algorithms can now be integrated in Python, a script-based programming language.

is a novel perspective because no decisive grouping is performed during the reasoning process. Since no explicit grouping is made, we must define the probability of a group-level scenario on an individual basis. For example, "the probability of the group that person i belongs to is chasing the group of j is 0.3". Inference using such probabilistic grouping over time leads to more robust reasoning, in particular on complex group scenarios. Table 8 provides an overview of group scenarios recognized by our system.

Group structure analysis: We analyze both the static group structures (size, compactness) and their dynamic changes (such as formation, dispersion) over time. The *size* of a group that person i belongs to is estimated as the expected value of the number of *healthy* tracks j that i is connected with:

$$G_s(i) = \sum_{\forall j} p_c^{\pi}(i,j)h(j), \tag{49}$$

where $h(j)$ is the track healthiness incorporated to deal with false and miss detections, by considering Kalman filter covariance and track lifetime.

We consider three status of a group structure: *(i)* the group is growing (formation), *(ii)* shrinking (dispersion), and *(iii)* remaining the same size (stable), with equations given in Table 8. The idea is to check all the neighbors of person i in $\mathbf{G}$ and see if there is a change in the connectivity. For example, if $\forall_{j \neq i}$, the group connectivity $p_c^{\pi}(i,j)$ is high at current time t and low at some previous time $t_p = t - T_w$, the probability of group formation of person i is high. We use a time window of

Table 4: Probabilistic group-level scenario recognition.

Group scenario	**Probabilities for track i, or between tracks (i,j)**
Group formation	$p_g^f(i) = \text{sigmoid}(y_g^f, 1, 0.2),\ y_g^f = \sum_{\forall j \neq i} p_c^\pi(i,j;t) \cdot \left[1 - p_c^\pi(i,j;t_p)\right] \cdot \max(h(i), h(j))$
Group dispersion	$p_g^d(i) = \text{sigmoid}(y_g^d, 1, 0.2),\ y_g^d = \sum_{\forall j \neq i} p_c^\pi(i,j;t_p) \cdot \left[1 - p_c^\pi(i,j;t)\right] \cdot \max(h(i), h(j))$
Stable group	$p_g^s(i) = 1 - p_g^f(i) - p_g^d(i)$
Loitering group	$p_g^l(i) = 1 - \prod_{\forall j} \left\{1 - p_c^\pi(i,j) p^l(j)\right\}$
Stable loitering group	$p_g^{sl}(i) = p_g^s(i) p_g^l(i)$
Distinct groups	$p_g^\delta(i,j) = \prod_{\forall k} \{1 - \max(p_c^\pi(i,k)p_c^\pi(k,j), p_c^\pi(j,k)p_c^\pi(k,i))\}$
Close-by groups	$p_g^c(i,j;t) = 1 - \sum_{k \neq i,j} \left[1 - p^c(i,k;t)\right] \cdot \left[1 - p^c(k,j;t)\right]$
Group meeting	$p_g^{meet}(i,j) = 1 - \prod_{t=t_0 \text{ to } t_f} \left\{1 - p_g^c(i,j;t)\right\}$
Group following	$p_g^{flw}(i,j) = p_g^\delta(i,j) \cdot \left[1 - \prod_k \left\{p_g^\delta(i,k) + [1 - p_g^\delta(i,k)] \cdot [1 - p^{flw}(k,j)]\right\}\right]$
Group chasing	$p_g^{chs}(i,j) = p_g^\delta(i,j) \cdot \left[1 - \prod_k \left\{p_g^\delta(i,k) + [1 - p_g^\delta(i,k)] \cdot [1 - p^{chs}(k,j)]\right\}\right]$

$T_w = 30$ frames.

Group scenario recognition: In security, group loitering is of particular interest to municipalities, because it is likely related to (or often the prologue of) illegal activities *e.g.* gang activities and disorderly youth. Our analytical definition of a loitering person has three criteria: *(i)* is currently moving slowly, *(ii)* has been close to the current position at a point in time in the past that was at least T_{min} seconds ago and at most T_{max} seconds ago, and *(iii)* was also moving slowly at that previous point in time.

For each person i, the probability of the belonging group G_i is loitering is one minus the probability that all other individuals in the group are not loitering. This *inversion technique* will be used frequently in subsequent group scenario analysis. An attractive characteristic of our framework is its flexibility to recognize new scenarios by combining existing knowledge. As an example, we detect a *stable loitering group* by multiplying the probability of stable group and group loitering (Table 8). Throughout the paper we denote subscript g as group level probabilities. [3]

[3] Although intuitively a group satisfying loitering should be stable in a larger time scale, each individual of it could

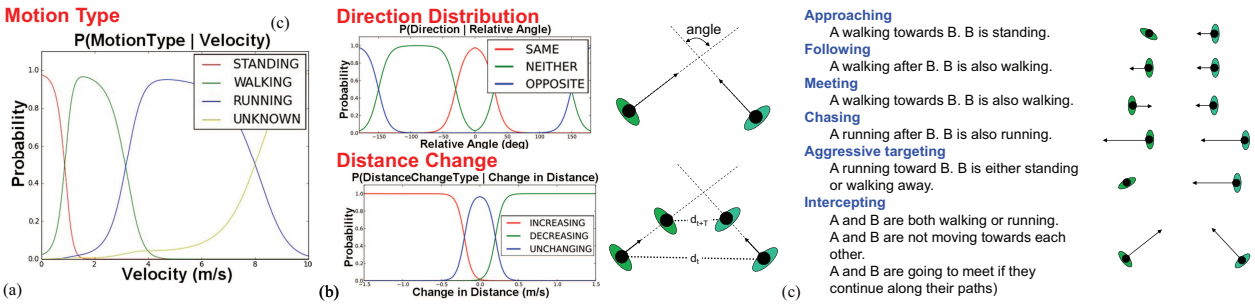


Figure 35: (a) **motion type:** The probability of a person being in different motion types is modeled by a set of sigmoid functions using velocity. (b, top) **relative motion direction:** The probability of different relative moving directions between two people is modeled by a set of sigmoid functions using relative angles. (b, bottom) **relative distance change:** The probability of different relative distance changes between two people is modeled by a set of sigmoid functions using the change in inter-person distance. (c) **pairwise interaction:** Illustration of different types of interaction between two people.

Pairwise group structure analysis: We consider two basic types of structure between groups: close-by groups and distinct groups. The former only considers pairwise group relationship at one time step while the latter considers track history. Two groups of a pair of people i and j are considered currently close-by if there exists no person k that both i and j are close to. Using the same inversion technique, we define the probability of close-by groups as: $p_g^c(i,j;t) = 1 - \forall_{k\neq i,j} p\big(k \text{ far from } i \text{ and } k \text{ far from } j \text{ at time } t\big)$.

The distinct groups p_g^δ between individuals i and j is modeled as $\{\forall_k$, there is no connectivity from (i,k) or $(k,j)\}$, using the same inversion technique. In addition, higher-level scenarios such as *stable distinct groups* and *stable loitering distinct groups* can respectively be defined as multiplications of component probabilities:

$$p_g^{s\delta}(i,j) = p_g^s(i)p_g^s(j)p_g^\delta(i,j), \tag{50}$$

$$p_g^{sl\delta}(i,j) = p_g^s(i)p_g^l(i)p_g^s(j)p_g^l(j)p_g^\delta(i,j). \tag{51}$$

Pairwise group scenario recognition: Building upon various per-track basis motion analysis for individuals such as meeting, following, and chasing, we can again recognize group-level scenarios using the soft group representation. The probability of the two groups of a pair of people i

still be considered not in a stable group in a smaller time scale. We multiply factors by assuming these are independent.

and j meeting at time t in the future is defined as:

$$p_g^{meet}(i,j) = 1 - \prod_{t=t_0 \text{ to } t_f} \{1 - p_g^c(i,j;t)\}, \tag{52}$$

where t_0 is the current time, t_f is the time extent in the future, and $p_g^c(i,j;t)$ is the probability of close-by groups.

We define the probability of a group G_i (where person i belongs to) follows an individual j as:

$$p_{gi}^{flw}(i,j) = p_g^{\delta}(i,j) \cdot (1 - p_{nf}(i,j)), \tag{53}$$

where p_{nf}, the probability of G_i not following individual j is defined as

$$p_{nf}(i,j) = \prod_k \big(p_g^{\delta}(i,k) + (1 - p_g^{\delta}(i,k)) \cdot (1 - p^{flw}(k,j))\big).$$

The intuition is that we consider each individual independently, taking account of two cases: either individual k and follower i are in different groups, or the k and i are in one group but k is not following j.

Next we use Eq. 94 to model the case where a group of individuals is following another group:

$$p_g^{flw}(i,j) = p_g^{\delta}(i,j) \cdot (1 - p'_{nf}(i,j)), \tag{54}$$

where the probability of G_i not following G_j is defined similarly as before, by taking account of two cases: either individual k and j are in different groups, or k and j are in the same group but k is not followed by i:

$$p'_{nf}(i,j) = \prod_k \big(p_g^{\delta}(j,k) + (1 - p_g^{\delta}(j,k)) \cdot (1 - p_{gi}^{flw}(i,k))\big).$$

The group-level chasing scenario can be defined similarly, in that the probability p^{chs} of chasing individual should be used. We further define a family of related group-level scenarios such as

group approaching, *group aggressive targeting*, and *group intercepting*, respective to the pairwise interaction outlined in Fig.92(c) in a similar fashion.

Detailed and formal exposition of scenario detection using the probabilistic group representation is in Appendix E.

8.4 Flanking Detection

The event of ***flanking***, or flanking maneuver (groups surrounding another group prior to an attack) is aimed at detecting a certain spatiotemporal configurations that is exhibited by groups before they engage in aggressive behaviors (see Figure 36). Data seems to indicate that an aggressive and dominating (in terms of strength and numbers) group tends to "surround" the victim group or individual or at least spatially spread out before the event. We consider the flanking condition in a probabilistic formulation as follows. Specifically, an individual T_k is flanked by two others T_i and T_j if:

1. T_i and T_j are in the same group.
2. T_i and T_k are in different groups.
3. T_j and T_k are in different groups.
4. The distance $d(T_i, T_j)$ is larger than $d(T_i, T_k)$ and $d(T_j, T_k)$, modeled by a sigmoid as in Eq.(55).
5. The angle θ_f between $\overrightarrow{T_k T_i}$ and $\overrightarrow{T_k T_j}$ is large enough, again modeled by a sigmoid in Eq.(55).

Specifically,

$$
\begin{aligned}
p(T_k \text{ is flanked by } T_i, T_j) = \; & p(T_i, T_j \text{ is in the same group}) \cdot && (55)\\
& (1 - p(T_i, T_k \text{ is in the same group})) \cdot && (56)\\
& (1 - p(T_j, T_k \text{ is in the same group})) \cdot && (57)\\
& \text{sigmoid}(d_{ratio}, \mu_{d_r}, \sigma_{d_r}) \cdot && (58)\\
& \text{sigmoid}(\theta, \mu_\theta, \sigma_\theta), && (59)
\end{aligned}
$$

where $\mu_\theta = 60$ and $\sigma_\theta = 20$ control how wide should the attackers T_i and T_j spread in order to flank the victim T_k; the distance radio $d_{ratio} = \frac{2d(T_i,T_j)}{d(T_i,T_k)+d(T_j,T_k)}$ controls the proper distance between the attackers and the victim; we use $\mu_{d_r} = 1.1$ (meter) and $\sigma_{d_r} = 0.2$ (meter).

We consider all pairs of T_i and T_j for every individual T_k in accumulating all probabilities as follows:

$$p(T_k \text{ is flanked}) = 1 - p(T_k \text{ is not flanked}) \tag{60a}$$

$$= 1 - \prod_{\text{all pairs of } T_i,T_j} p(T_k \text{ is not flanked by } T_i, T_j) \tag{60b}$$

$$= 1 - \prod_{\text{all pairs of } T_i,T_j} (1 - p(T_k \text{ is flanked by } T_i, T_j)) \tag{60c}$$

Figure 36 illustrates example flanking events detected in our system.

8.5 Contraband Handoff Detection

A more complex example is that of "Contraband Handoff" between two inmates (see Mock Prison Riot 2010 data collection described later in this document) that can verbally be described as follows:

> A contraband handoff at time t requires two individuals to be physically close to each other during handoff. Two seconds before the handoff, the individuals are not close to each other but approach / move toward each other. At least one person is walking during the approach (i.e., the second person might be standing or himself be walking).

The scenario can be described as:

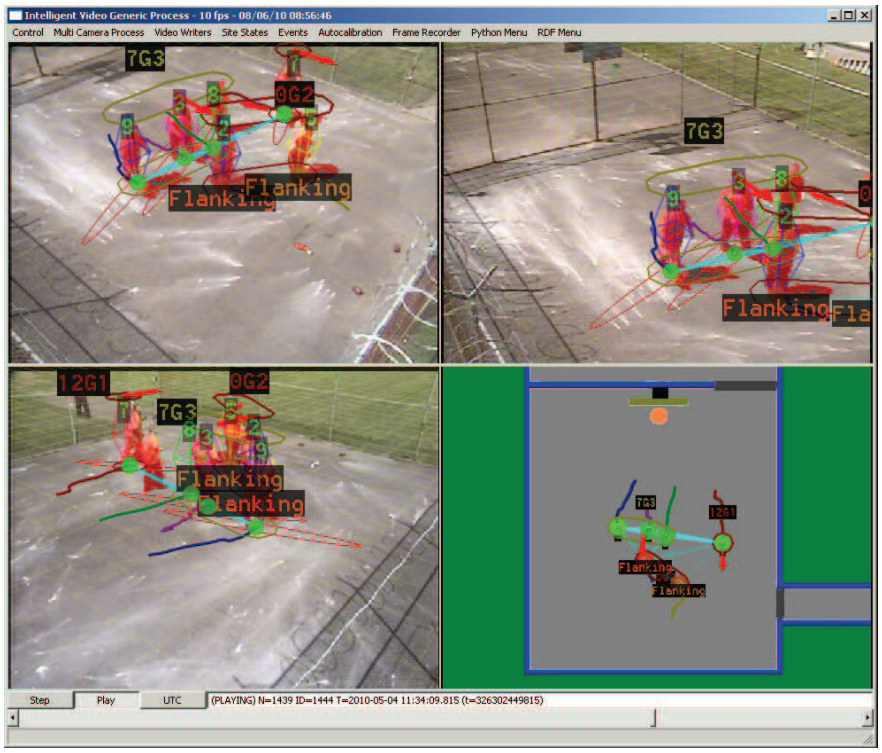


Figure 36: **Probabilistic Flanking.** This figure illustrates a flanking event where a group of 4 attackers surrounds two other victims. The calculated probabilities of each individual being flanked is visualized by a color circle ranging from green (0, unlikely) to be red (1, likely). Observe how effective our proposed probabilistic framework in capturing the two surrounded victims at the center. The shown activities have been enacted by law enforcement and corrections personnel.

$$P(\{\text{A and B are exchanging contraband}\}|A,B,t) = \tag{61}$$

$$[P_{\text{motion}}(D=\text{WALKING}|A,t_1) + P_{\text{motion}}(D=\text{WALKING}|B,t_1) - \tag{62}$$

$$P_{\text{motion}}(D=\text{WALKING}|A,t_1)P_{\text{motion}}(D=\text{WALKING}|B,t_1)] \cdot \tag{63}$$

$$P_{\text{meet}}(C=\text{True}|A,B,t_1)P_{\text{dir}}(D=\text{OPPOSITE}|A,B,t_1) \cdot \tag{64}$$

$$P_{\text{close}}(P=\text{False}|A,B,t_1)P_{\text{close}}(P=\text{True}|A,B,t), \tag{65}$$

where $t_1 = t - 2.0$ sec.

8.6 Event Triggering and Modeling

Once a probabilistic estimation of a scenrio is computed, the last step is to determine whether or not an alert to the user should be triggered. Since an exhausted ground truthing is not feasible in our case, typical data-driven or learning based approach does not apply. We follow a standard Receiver Operating Characteristic (ROC) analysis [5, 6] to determine a best threshold for event triggering.

Since no ground truth is available, our event triggering analysis is based on an assumption that the event probability estimation is accurate and confident. Given a video with a sequence of event probability estimation over time as shown in Figure 37, the goal is to determine a good threshold t. Note that such a threshold explicitly imposing a corresponding *box kernel* to the probability signal, depicted as the dotted curve in Figure 37, which essentially ditates the signal probability above t should be treated as probability 1 (and thus being triggered, green in Figure 37), the signal probability below t should be treated as probability 0 (and thus not being triggered, orange in Figure 37). Based on such interpretation, the outcome of the binary prediction in the 4 cases, namely, the true positive (TP), false positive (FP), true negative (TN), false positive (FP) can be defined (which is also known as a confusion matrix). The ROC curve is a plot of true positive rate (TPR) against false positive rate (FPR):

$$TPR = \frac{TP}{TP + FN} \tag{66}$$

$$FPR = \frac{FP}{FP + TN} \tag{67}$$

Each threshold t generates a confusion matrix which constitute a point on the ROC curve.

The upper left corner in a ROC chart represents the ideal performance (where no false alarm is generated with no miss-detection). So an intuitive solution to select the best threshold is to determine the point where the ROC curve is closest to the ideal corner. However, the determination of such "optimal" threshold is often domain specific, since how much the false alarm rate could be tolerated against the trade-off between higher detection rate is application dependent. Specifically,

one might choose a different strategy to pick a point out of the ROC curve to represent such best solution. Typical solution involves finding the 45 deg tangent line closest to the ideal corner, where the tangent point gives the best threshold. One can variate the slope to produce a family of lines which favors different weights to trade-off between TPR and FPR. Other solution include finding the max of $TPR \cdot (1 - FPR)$ [27]. Figure 37 depicts yet another solution, to defind the optimal threshold as the closest point of the ROC curve to the ideal corner.

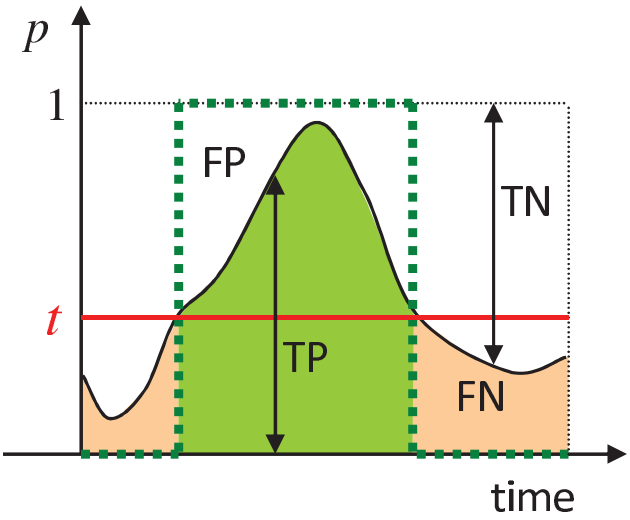


Figure 37: **Receiver Operating Characteristic (ROC) analysis for automatic determination of the event triggering threshold.** See text for explanation.

The above ROC analysis provides a basis for selection and tuning of parameters which we use to automatically determine a threshold for event triggering. Our approach only requires to run the experiment on a few videos containing the representative scenario of interest. Based on the resulting event triggering rate such as the one shown in Figure 38, the operator can quickly adjust a proper parameter setting to trade off the desired false alarm and mis-detection rates.

In addition to the ROC analysis, our other strategy is model event duration so as to avoid repeating triggers of the same event. To illustrate, a loitering event is likely to be observed for a long duration of time, and we certainly do not want the system to alarm the user for every instance of detection. We model event duration by using an *armory mechanism*, that is, once an event is triggered, any subsequent event detected during a period of time is kept silent and only used to update (extend) the period of time. In other words, the same event will be triggered only after it is

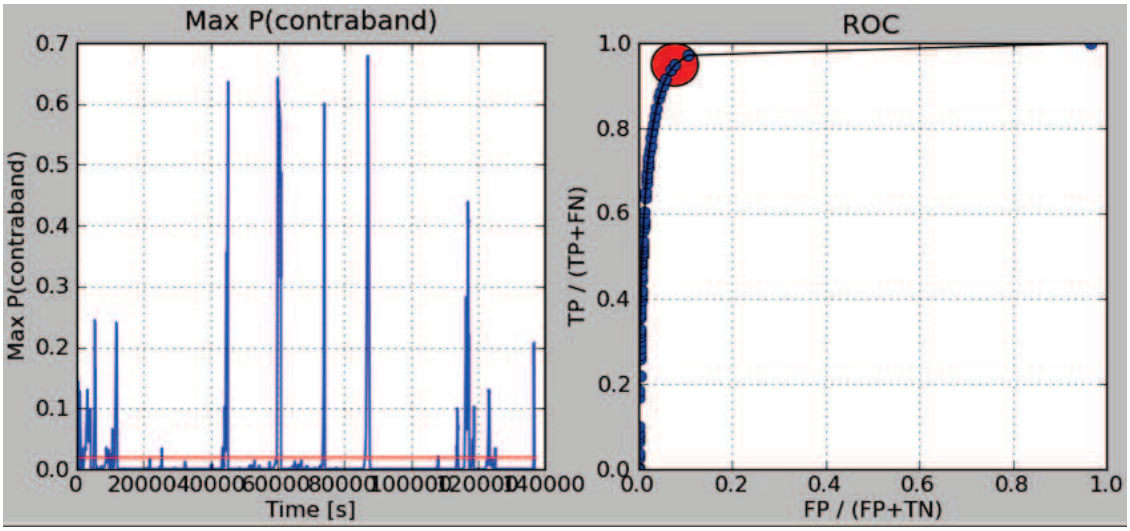


Figure 38: **Automatic determination of event triggering threshold.** (Left) The estimated probability of the contraband handling event over time, computed from a video clip containing several contraband handling scenarios. (Right) The ROC analysis suggests a theoretically optimal threshold t_e for event triggering. Here $t_e = 0.03$ and is depicted as a red line on the left. Observe that major contraband instances with high probability are automatically selected by this threshold, while noise and instability are filtered out.

not detected for some pre-defined duration of time. We found this mechanism effective in filtering out unwanted events in practice. Figure 66 shows a typical event table generated by our system.

Realtime Scenario Recognition: So far, we have described the modeling of various low and mid level tasks using Bayesian Networks. For practical purposes, it is necessary for the system to make a final decision at some point. We want to do this while avoiding compromising the robustness and elegance of the soft assignment paradigm. It is thus undesirable that one should make a final decision with some sort of hard thresholding.

Instead, we leverage the power of Bayesian modeling by enhancing it with what is known as **decision** and **utility** nodes. A decision node describes a decision, and contains mutually exclusive decision states. So, for example in the fast person detection model, a decision node would basically be comprised of a "yes" and "no" state. Such a decision node is not assigned any probabilities. Rather, the outcome of the Bayesian inference populates each decision state with a probability score, allowing one to compare and select a decision state that is relatively most probable.

To influence the Bayesian inference on a decision node, one would typically utilize a utility

node that encapsulates the utilities associated with all possible outcomes. To illustrate, we can refer to Figure 22, whereby the node labeled "Satisfaction" is an utility node. In it, as an example, we specify that if "IsTrueDetection" is true, "IsPerson" is true, and "IssueTrueDetection" is "yes", then a high utility score should be given. A decision node (in the figure, "IssueTrueDetection") attached to such an utility node completes the influence diagram.

8.7 Event Explanation

Since our inference engine is probabilistic (Bayesian), and is highly modular, the explanation of its triggered events is *explicit* from the reasoning process — the probability in reasoning can provide straightforward verbal explanation. For example, the explanation of a loitering event could be: "The loitering event is detected for target 10 with $P(loiter) = 71\%$ because: (1) the target is currently moving slow with $P(slow) = 86\%$, (2) the target has been close to its current position at a point in time in the past within a window of 10.0 and 20.0 seconds ago, and (3) the target was moving slow at that previous point in time with $P(slow) = 33\%$." Refer to Figure 39 for an example of the explanation of a detected "Stable loitering group"event. Note that **backtracking of explanation** is possible. Since the "Stable loitering group" event is based on two components, the "Stable group" and the "Loiter group", the explanation could be backtrack into each of the ancestor reasoning nodes, where the explanation of "Loiter group" can be traced into the explanation of the "Loitering" event. Continue from the aforementioned loitering example, the system can further explain how the person was determined to be slow with $P(slow) = 33\%$ in the past.

Figure 40 illustrates another example of the explanation of a detected "Contraband Handoff" event. Recall from previous report that this event is detected based on five reasoning nodes:

$$
\begin{aligned}
& p(\{i \text{ and } j \text{ are exchanging contraband at time } t\}) \qquad (68)\\
=&[p^{mt}(\textit{walking}|i; t_p = t - T) + p^{mt}(\textit{walking}|j; t_p) - p^{mt}(\textit{walking}|i; t_p)p^{mt}(\textit{walking}|j; t_p)]\cdot\\
& p^{meet}(i, j; t_p)p^{md}(\textit{opposite}|i, j, t_p)\big[1 - p^{c}(i, j; t_p)\big]p^{c}(i, j; t).
\end{aligned}
$$

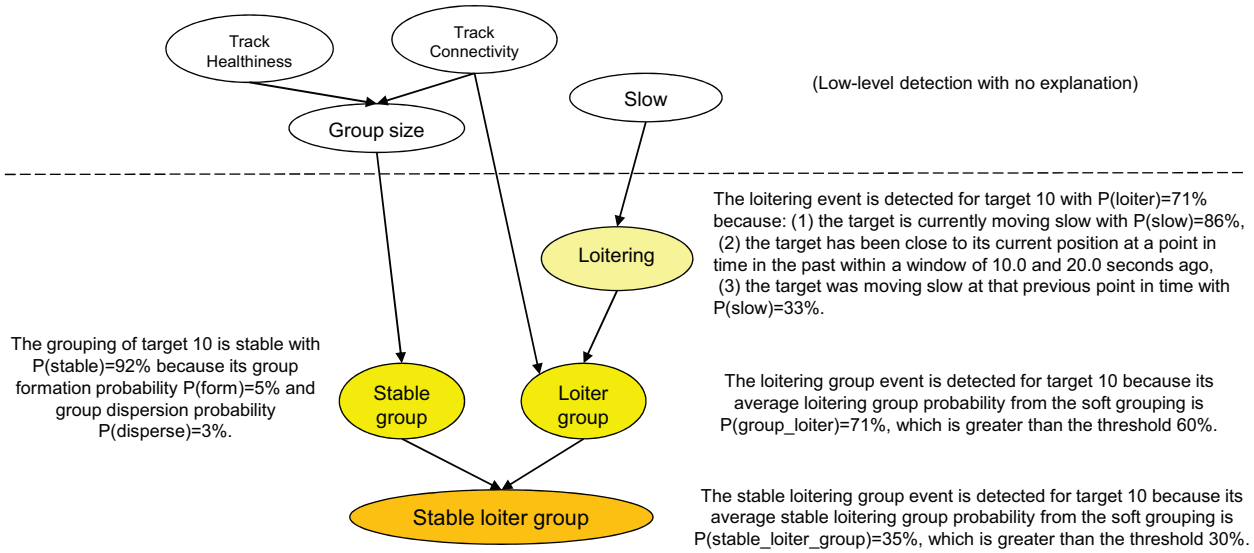


Figure 39: **Event Explanation** and backtracking of the explanation of a detected "Stable Loitering Group" event.

where p^{mt} is the motion type probability, p^{meet} is the probability of the meeting of two individuals, p^{md} is the motion direction probability, p^{c} is the probability two individuals being close. Observe that the final triggering probability (31%) can be backtracked and decomposed into the five terms of ancestor probabilities $(= 97\% \times 59\% \times 65\% \times 100\% \times 82\%)$.

The backtracking of explanation is simple and transparent. It also reflects basic system characteristics and robustness. For example, in case the operator wants to fine tune the system parameters to emphasize a particular component (*e.g.* to better incorporate domain knowledge), consistent patterns of the explanations provides a baseline to improve the overall parameter configuration.

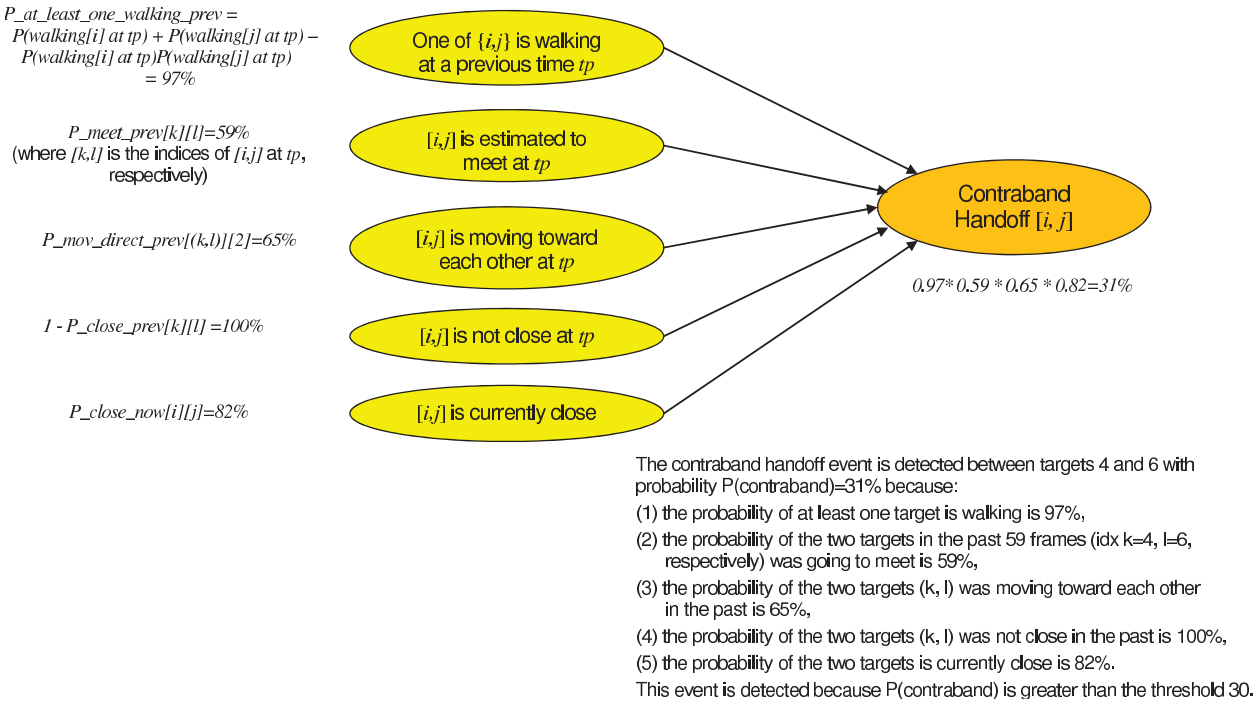


Figure 40: **Backtracking of the explanation** of a detected "Contraband Handoff" event.

9 Scenario Modeling GUI

To enable operators to quickly create models that recognize domain-specific scenarios, we have developed a *visual programming* framework that represents scenarios by a flow of information through a network of processing steps. This approach is motivated by the proliferation of graphical models [7] in general and Bayesian Networks [8] specifically for recognition. In visual programming, algorithms and computational procedures are represented by nodes connected by directional edges. For a given node, its incident edges represent incoming data and exiting edges represent the data produced by this node. Visual programming paradigms have emerged from many different applications such as the programming of toy robots (Figure 41) and industrial measurement and simulation systems (Figure 42).

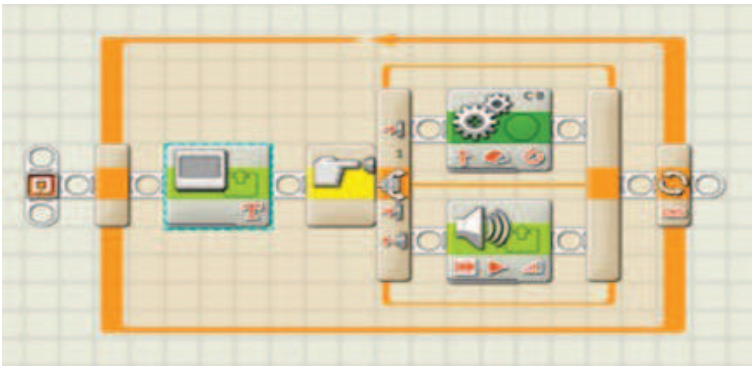


Figure 41: **Visual Program for Toy Robot.** This example is for the LEGO Mindstorm platform.

In visual programming, algorithms and computational procedures are represented by nodes connected by directional edges. For a given node incident edges represent incoming data and exiting edges represent the data produced by a node. The example in Figure 43 represents an abstract example applicable to our problem domain. It represents the step of finding the latest `frame_information` entry in the resource database, in other words the current frame information for a live running program. The center node called "find RDF array" has two input ports: (i) pointer to the RDF tree and (ii) a string denoting the name of the entry to be retrieved from the RDF database. The output of this node is an array of RDF trees. The right-most node represents the operation of finding the last entry in this array. It produces a pointer to the RDF subtree that is

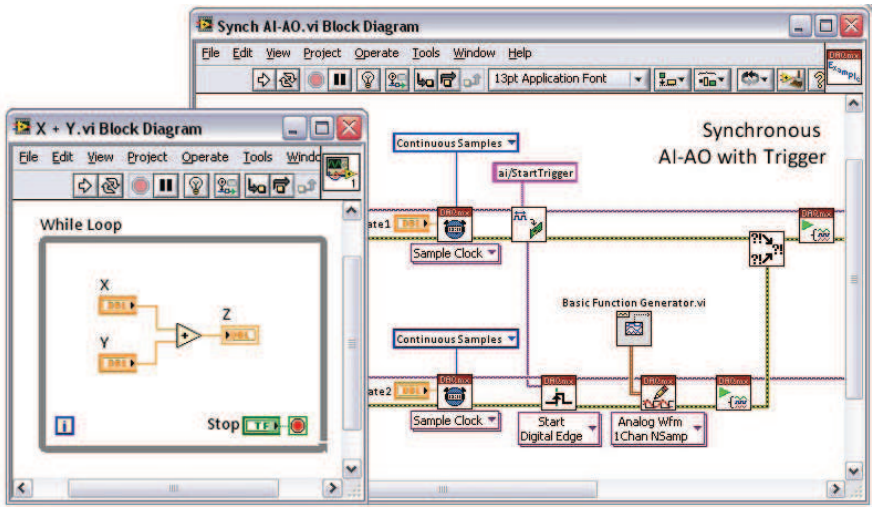


Figure 42: **Visual Program for Industrial Equipment.** This example is from National Instruments LabVIEW platform `http://www.ni.com/images/coreblock/large/fasterprogramming.jpg`.

contained at this location.

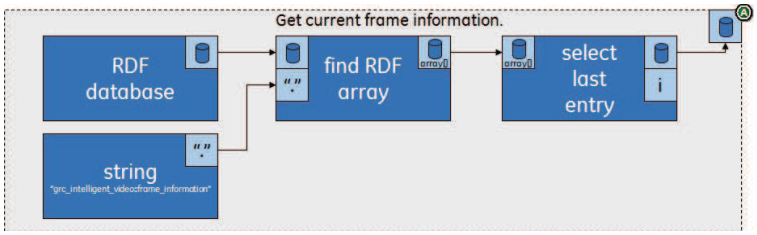


Figure 43: **Resource Database Lookup as Visual Program.** See text for details.

Similar to the trivial example above, more complex algorithms can be developed. Also, nodes can take the role of higher level processes. Figure 44 illustrates the scenario modeling GUI that we created for the recognition of "stable loitering distinct groups" — to identify if there exist two distinct groups in the scene that are both stable (no individuals joining or leaving the groups) and are currently loitering. Such distinct groups are useful in suggesting a possible group conflict before it takes place. The developed scenario modeling GUI allows operators to easily define new scenarios using a pre-defined bank of event inference modules in the form of process nodes, which serves as building blocks that can be drag-n-dropped and connected to create new scenarios. The process nodes are categorized into several functional groups, including RDF accessing nodes, inference nodes, logical nodes, event triggering nodes, and visualization nodes. Users can visually

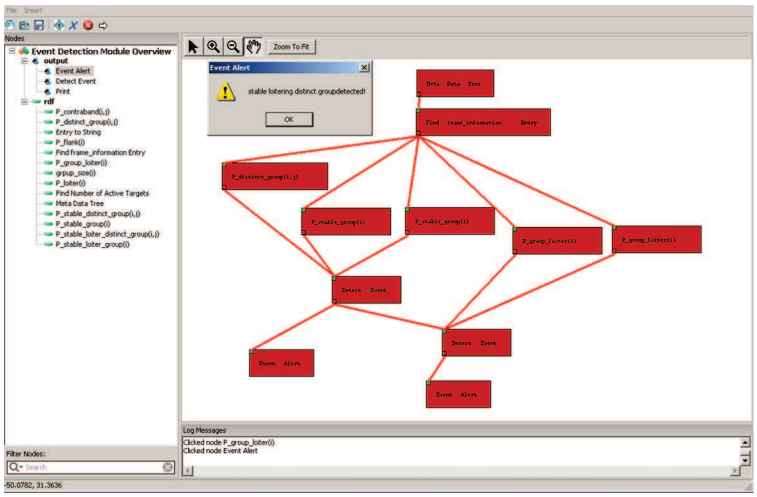


Figure 44: **Scenario modeling GUI** for detection of event "stable loitering distinct groups".

link and combine these modules following the visual programming paradigm to create customized scenarios, without the burden of developing cumbersome computer codes.

10 Advanced Scenario Recognition

The scenario recognition approach covered so far can be viewed in the *rule based* category, which requires manual specification of pre-defined events. In this section we describe scenario recognition methods that are more sophicated. By leveraging recent developments in the machine learning and artificial intelligence communities, we have investigated two separate but closely related methods, namely the *leanring based* approach (detailed in Section 10.1) and *symbolic logical reasoning* approach (detailed in Section 10.2).

10.1 Learning-based Approach

Through ongoing and previous NIJ programs, we have developed technologies capable of high-level event modeling and recognition based on estimated low-level evidence. To enhance the low-level event recognition module with advanced learning based methods, we have developed a machine learning based event classification method. Specifically, spatial temporal histograms and robust higher order features are extracted from training sequences and clustered using *e.g.* bag of visual words, a widely used technique in computer vision. Support vector machine based classifiers are then trained on these features to classify events of different categories.

10.1.1 Introduction

We have successfully applied the rule-based method for detecting the events. However, there are two main disadvantages for the rule-based method: 1) we need to manually define a rule each time we have a new event category; 2) the detection performance highly depends on how good the defined rules are. The defined rules may not be discriminative enough to recognize different event categories. To solve these problems, we develop machine-learning based methods to automatically learn the event detectors. As illustrated in Figure 45, during training, we are provided some video segments labeled with the correct event categories. The detectors will be automatically learned with these labeled data. During testing, given a new video/video segment, the detectors will decide

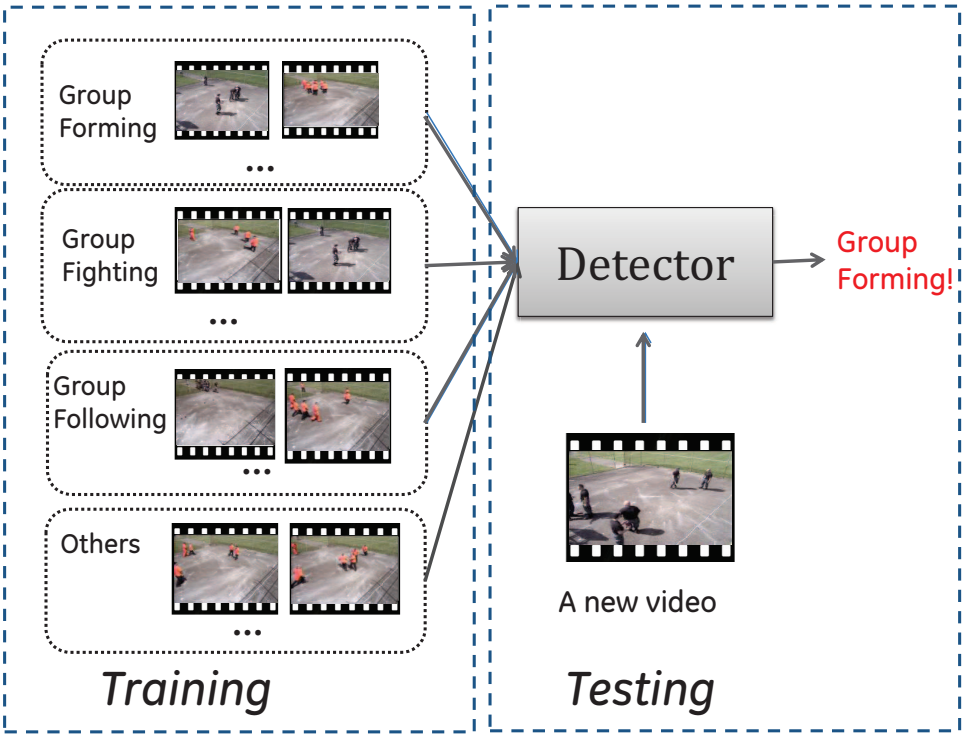


Figure 45: The overview of the learning based event detection system.

whether the events of interest occurred in it.

10.1.2 Approach

The overview of the approach is as follows. We first extract feature histograms from the frame of the videos. Then we cluster the histograms into words, and model the temporal changes of the words. Finally, we perform classification of the videos using a SVM classifier with a proposed kernel based on the temporal modeling of words. Figure 46 shows the steps for feature extraction and word generation.

Feature Extraction: First, we extract useful features from the videos. We assume that people in the video have already been detected with the person detector and tracker. For the event recognition task, we only use the location and velocity information of the persons, instead of the appearances of the videos. This type of information is more robust for different scenes and environments, and very fast to process. We extract four types of features as follows. For each feature

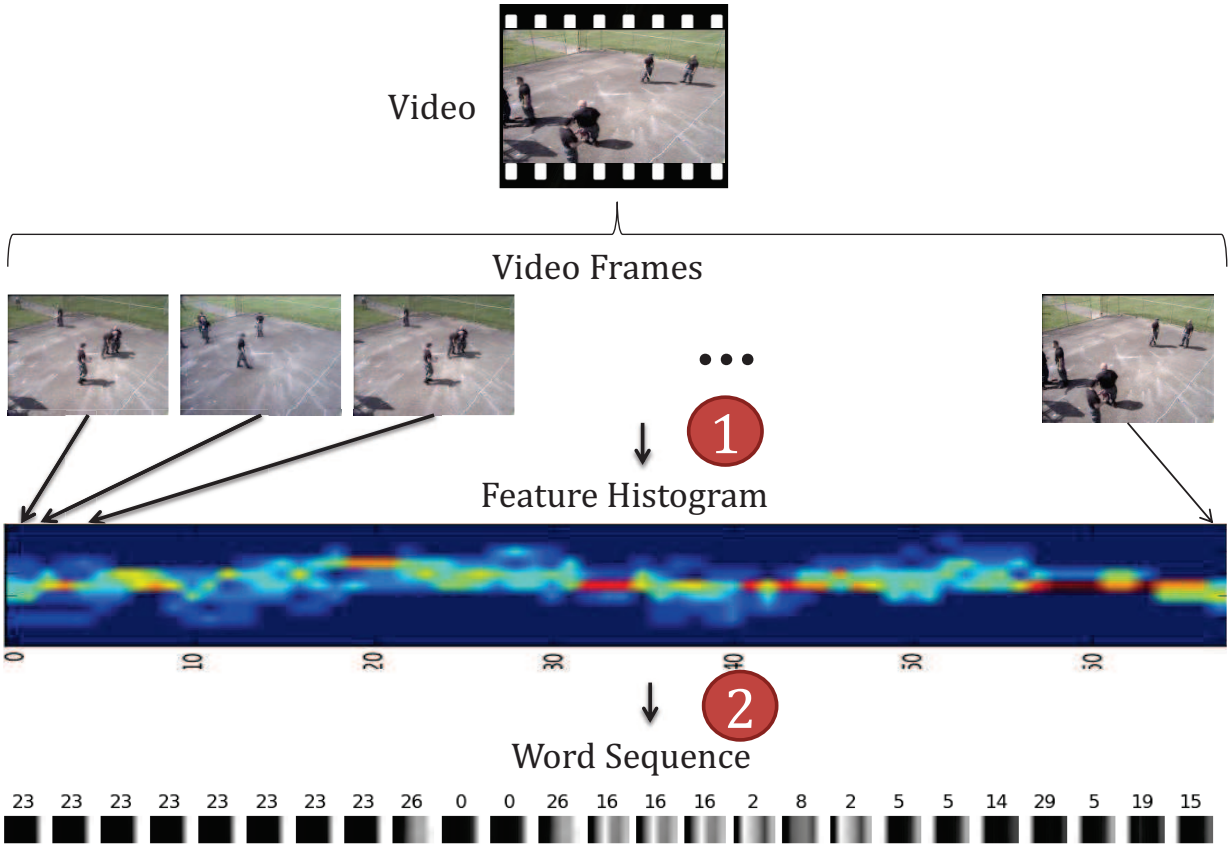


Figure 46: Approach pipeline. In the first step, we create a histogram for each frame in the input video. Each column in the histogram image represents the histogram for a frame. Red indicates larger values, and blue indicates smaller values. At the second step, we represents the histogram with words. Each word is created with the histogram of consecutive 4 frames. For the word appearance in the figure, each row is a histogram for a frame.

type, each frame of a video is represented a histogram of the feature values, as illustrated in figure 46 (Step 1).

1. **Connectivity:** The connectivity between two persons is defined as the likelihood that the two persons are in the same group. The likelihood is computed using our group analysis algorithm. For each video frame, we compute the connectivity for each pair of persons, and represent the frame as a histogram of the connectivity values. The histogram is created by discretizing the connectivity value (in $[0, 1]$) into N bins, and count the number of person pairs whose connectivity values are in each bin. Figure 47 shows the connectivity histograms of three example videos of different event categories. For the 'Group Dispersing' event, the connectivity values change from large to small, while for the 'Group Forming' event, the values change from small to large. The values for the 'Group Following' event remains large over time.

2. **Connectivity Change:** For each pair of persons, we compute the connectivity difference in the current frame and in the previous frame, if the two persons are also detected in the previous frame. Thus, each frame is represented as a histogram of the connectivity changes. The histogram is created by counting the number of person pairs whose connectivity changes are in each discretized bin.

3. **Speed:** We compute the speed of each person using the location differences between frames. The histogram of speed for a frame is calculated by counting the number of persons whose speeds are in each discretized bin.

4. **Moving Direction:** We also record the moving direction of each person in a frame by the angle of the location difference (in $[0, 2\pi]$). To deal with camera rotations, we normalize the angles of each person by subtracting the mean of them. The histogram of the angle is calculated with the normalized angle values.

Thus, for each feature type, the video is represented as a sequence of histograms. Events should be recognized by modeling the changes of the histogram values over time. Directly modeling

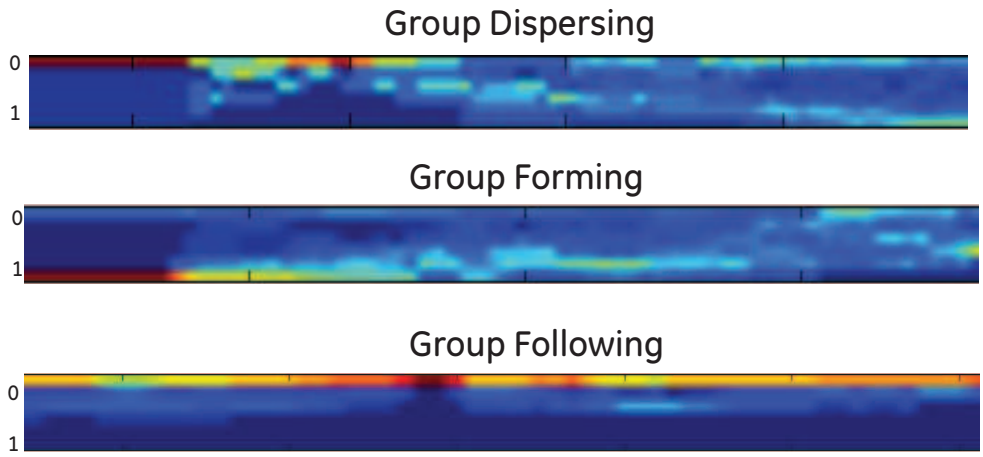


Figure 47: Connectivity histograms for example videos of three different events. Red indicates larger value, and blue indicates smaller value. Each collum is the histogram of a frame. From left to right, the figure shows the histograms of frames over time.

the temporal histogram differences is difficult. Therefore, we further cluster the histograms into "words".

Bag-of-Words Event Detection: To create the words, we first represent each frame by concatenating the histograms of the current and previous consecutive T frames. The concatenated histogram models the local histogram values. Then we cluster the concatenated histograms using K-means into |V| clusters. We call each cluster a "word". A word represents a certain pattern of the local histograms. Thus, for each feature type, a frame is represented with a word, and a video turns to be a sequence of words as in Figure 46 (Step 2).

After representing the videos with a sequence of words, we can use the Bag-of-Words (BoW) model for event recognition. The BoW model discards the location of the words, and represents the video by the histogram of the word occurrences. The length of the histogram is the size of the vocabulary, and the value for each bin is the occurrences of the corresponding word in the video. A general classifier, such as the Support Vector Machine (SVM), can be used to classify the word histograms. The BoW model is simple and efficient; however, it does not capture the temporal information of the words. Figure 48 illustrates the words of two example video sequences of event 'group forming' and 'group dispersing'. Without considering the temporal information of

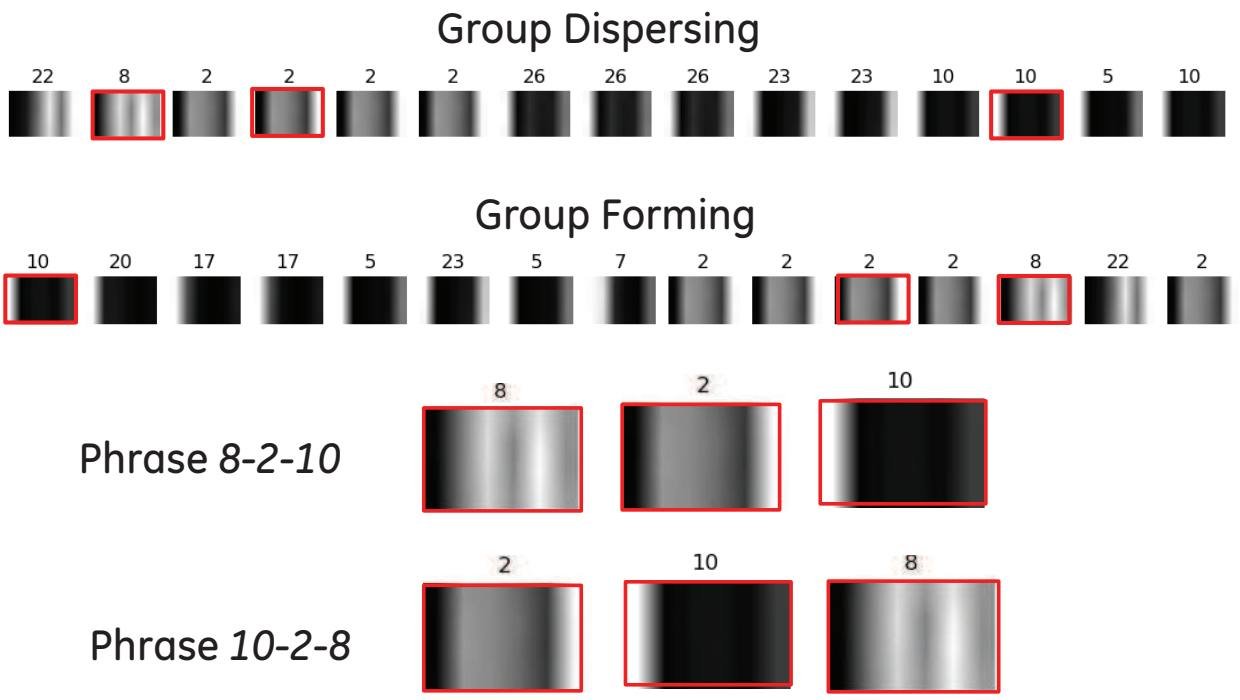


Figure 48: Word sequences of two example videos of event type “group forming" and “group dispersing".

the words, the word occurrences are quite similar for the two event types.

Temporal Modeling with Bag-of-Phrases:

As already described, a word will represent a certain pattern for the local feature histograms. To model the histogram changes over time, we can utilize the word changes. We use the notation of phrases to model the word changes. A phrase is a sequence of words in a particular order and intersections between them. A phrase can be constructed with non-continuous words, and thus is robust to behavior changes of the same event category. We also define phrases with words far from each other, so we can model the long-range temporal changes. Figure 48 shows two example phrases that occurred in the two videos. Phrases are more discriminate than words for classifying the two word sequences.

With the phrases, a video can be represented as a Bag-of-Phrases (BoP) histogram, whose length is the number of all possible phrases. However, the number of phrases increases exponentially as the number of words in one phrase increases. Thus, the computation time will become

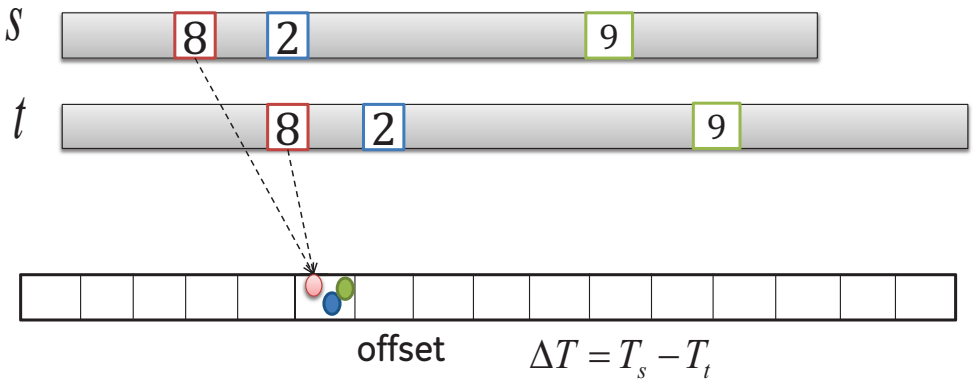


Figure 49: Illustration for the algorithm of finding co-occurring phrases.

expensive as we increase the length of the phrases. Similar to the algorithm proposed in [28], we propose a *correspondence transformation* algorithm that can model the phrases of any length in linear time.

The key idea is to find the co-occurring phrases in two sequences. As shown in figure 49, if two phrases in two sequences are composed of same N words, and the N words are in the same order and intervals among them, we have a co-occurring length-N phrase. Thus the algorithm for finding such phrases is as follows. For each pair of same word from the two sequences, t and s, we compute their offset, that is the time in t subtracts the time in s, and create a vote on the offset space at the corresponding location. After processing all same word pairs, if we have N words at the same location on the offset space, we have a length-N co-occurring phrase in the two sequences. Thus, to compute the total number of length-N phrases in the two sequences, we find the locations that have more than N votes on the offset space. If we have M_1, M_t votes at location $1, , t$, which are larger than N, the total number of co-occurring length-N phrases would be computed as follows.

$$K_N(s,t) = \sum_{i=1,..,t} \binom{M_i}{N} \tag{69}$$

To deal with the events with different durations, we scale the sequences with different factors. Specifically, we add a scale dimension on the offset space. A pair of same words will create a vote for each scale ds at offset $T_s - T_t \times ds$. Thus we can find co-occurring phrases that are composed of same words in the same order but in different intervals.

Classification with Phrase-Kernel: We classify the videos with their Bag-of-Phrase representations with SVM. Because of the kernel trick, SVM only requires the inner-product of the Bag-of-Phrase histograms between two videos for both training and testing. As shown in [28], we can prove that the number of co-occurring phrases of two videos equals to the inner product of Bag-of-phrase histograms of them. Therefore, we use the number of co-occurring phrases of two videos as a kernel for the SVM. As we have shown, this kernel value can be calculated in time linear to the number of words in one sequence for any phrase length.

Practically, during training, we compute the kernel matrix of the training data. The value at (i, j) in the kernel matrix is the number of co-occurring phrases for video i and j. Then we input the kernel matrix to Libsvm to learn the classifier. The classifier will give weights to the training data. Those that have weight 0 are called support vectors. During testing, we compute the kernel value of the testing video with the support vectors, and get the final decision value by summing over the weighted kernel values. Weights are the coefficients of the support vectors.

10.1.3 Results

We manually created 180 video segments for evaluation. Each video segment is labeled with the corresponding event category. If no interesting events occurred in a video, we label it as 'random'. We do not require clear start and end point for each event in the video segments. As long as an event occurred in the video, we label it with this event. This property is also an advantage of our algorithm. We randomly select 60% video segments for training, and the rest 40% for testing. Table 5 shows the number of videos for each category. Figure 50 shows the ROC curves for each category with BoW and BoP of different length phrases. According to the figure, we can already produce around 90% area under curve with the BoW representation, and BoP can further improve

Table 5: Number of video segments for experiment

Events	Group Dispersing	Group Following	Fighting	Random
Num. of Videos	22	16	11	135

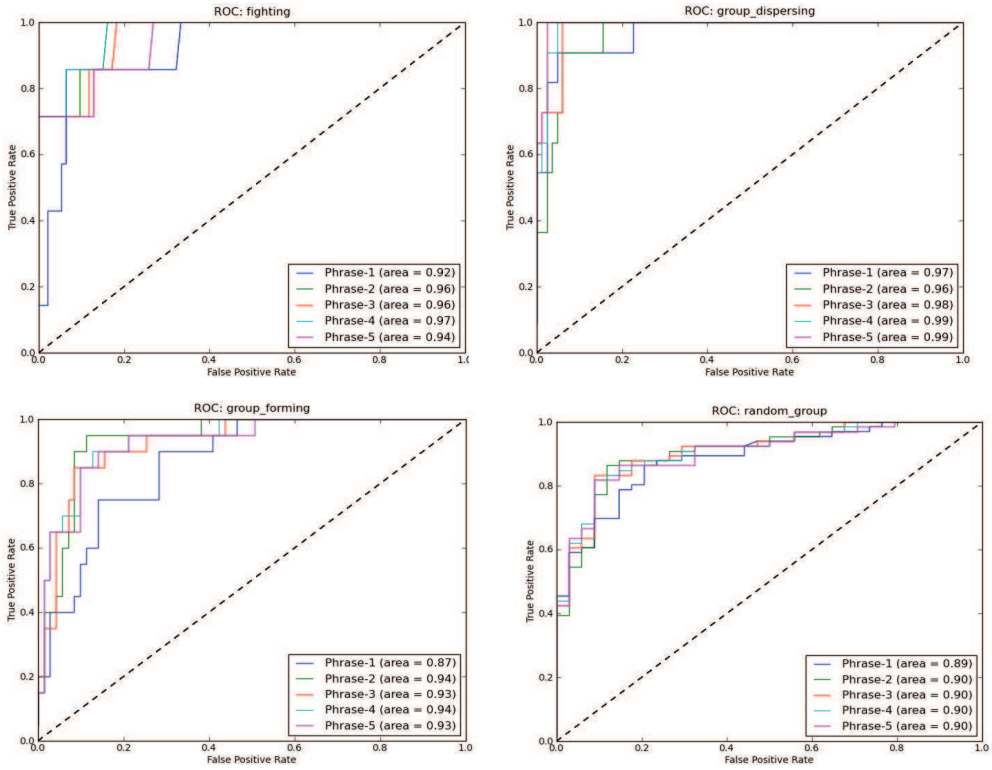


Figure 50: ROC curves for the four event categories.

the performance of BoW.

We also created a GUI to facilitate the learning and testing of the event detectors. Figure 51 shows a screen shot. The users can learn the detectors by loading the labeled video segments, and perform prediction of testing videos. The GUI will show the event detection results with confidence values. The user can also click the items to play the video segments of the detected events.

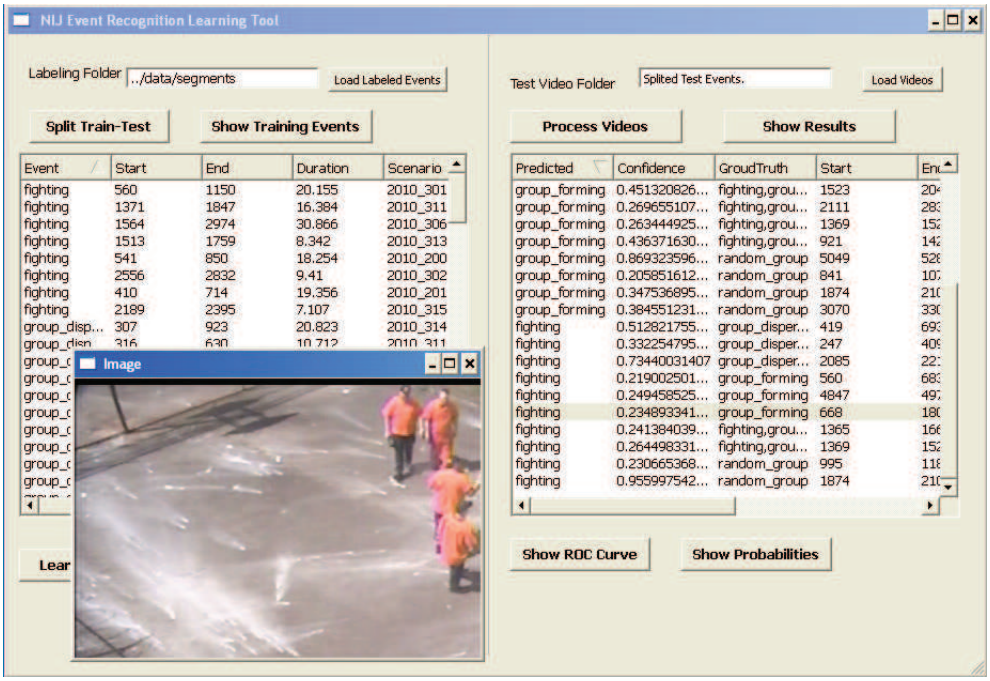


Figure 51: Screen shot of the User Interface.

10.1.4 Conclusion

We proposed a novel learning based framework for group-level event recognition. Unlike most existing event recognition works, which define the events based on the movements of an individual or the entire crowd, the events discussed in this paper focus more on the interactions among people. We designed robust features that can capture the group context of individuals in a video. We built a system with the proposed algorithm, which can process a video and detect the events in real-time. The performance of the system significantly outperforms the state-of-the-art method on a challenging dataset. More technical details about the approach can be found in Appendix G.

10.2 Symbolic Logical Reasoning Approach

Specifically, based on the low level video event detection modules, we can detect long temporal events by combining temporal logic deduction with probabilistic reasoning. The events are categorized as primitive events and complex events, of which the former are basic activities that can be

directly inferred from analysis of the video stream, and the latter correspond to complex activities that are defined with a temporal sequence of primitive events in a consistent manner using logical connectives such as AND, OR and NOT and temporal relations specified with temporal logic (e.g., Allen's interval algebra [29]). The primitive events are assigned with probabilities corresponding to the intrinsic uncertainties in the low level video event detection modules, and the complex events are assigned with probability that are chained from their defining primitive probabilities. In the actual deployment of the video event detection system, each event is implemented as a probabilistic filter that are plugged into the video streams as filters that will be triggered with a probability above a preset threshold.

In this part of the project, we aim to incorporate some abilities for temporal inference of events with relatively long durations into the current video event detection system. Specifically, we made progress in two directions. First, we developed a theoretical framework for temporal event detection based on temporal declarative logic. In this framework, we categorize temporal events of interest to the current system into primary events, which are events whose probabilities can be directly assigned from the tracking module, and complex events, which are events defined based on primary events using logic connectives (AND, OR and NOT), and whose probabilities are computed based on the probabilities of the defining primary events. We formalize the probability assignment procedure, which will form the theoretical basis of the subsequent development of long durational event detection.

We further focused on one particular type of complex event in the current system we call as persistent events, which are event types that persist across a temporal interval, and intuitively correspond to actions that can be described as "have been doing something for certain time". In particular, we define the probability of temporal persistent events as a temporal smoothing of the probabilities of the underlying primary event types. This is a building block of the subsequent system module that implements detection and inference of long durational events.

11 Advanced Gaze Tracking

We present a comprehensive approach to track one or more individuals' gaze directions by estimating their locations, body poses, and head poses in an unconstrained environment. The approach combines person detections from fixed cameras with directional face detections obtained from actively controlled pan tilt zoom (PTZ) cameras. The main contribution of this work is to estimate both body pose and head pose (gaze) direction independently from motion direction, using a combination of sequential Monte Carlo Filtering and MCMC sampling. There are numerous benefits in tracking body pose and gaze in surveillance. It allows to track people's focus of attention, can optimize the control of active cameras for biometric face capture, and can provide better interaction metrics between pairs of people. The availability of gaze and face detection information also improves localization and data association for tracking in crowded environments. The performance of the system will be demonstrated on data captured at a real-time surveillance site.

Detailed and formal exposition of this topic is presented in Appendix F "Advanced Gaze Tracking". This technical paper won the Best Paper (Runner Up) Award at IEEE's premium international conference on video surveillance, the "IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), which took place at Klagenfurt, Austria on Aug. 2011.

12 System Deployment and Evaluation

Our video analytic system has been developed mainly at the GE Global Research Center's test site. The system has been later deployed and evaluated at various other sites. The GE GRC test site consists of 4 calibrated fixed cameras and 4 pan-tilt-zoom (PTZ) cameras looking at a courtyard. The cameras are mounted at about 3 meters height looking roughly about 30 degrees downward to the courtyard, as shown in Figure 52. The camera resolution is 640x480 and can operate in 320x240 to speed up the process if many views are used. The framerate for behavior tracking of a group of 3 to 10 individuals using 4 fixed cameras is about 15 frames-per-second (FPS), depending on how many pedestrians are in the view. The FPS for gaze tracking using 4 fixed and 4 PTZ cameras for 3 individuals is about 5 to 10 FPS.

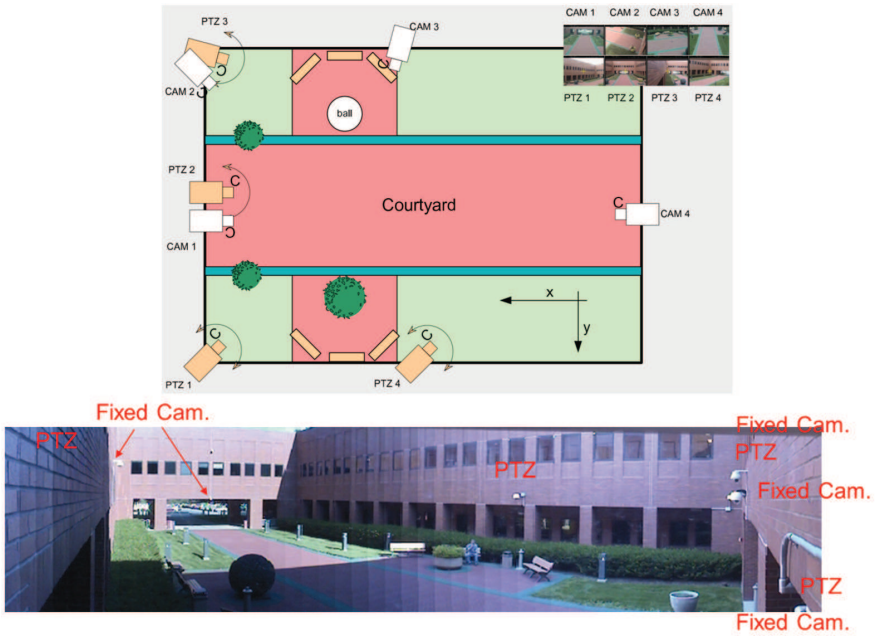


Figure 52: Camera setup at the GE Global Research Center courtyard test site.

In the following sections, we will describe the system deployed and evaluated at (1) the MPR 2010 test site, and (2) a newly constructed real-life street site through a collaboration with a local police department. The system will also be independently installed and configured by ManTech Inc. at their own sites for a full evaluation on behalf of NIJ.

12.1 System Evaluation in MPR 2010

The West Virginia High Technology Consortium (WVHTC) Foundation's Mock Prison Riot (MPR) facilitated an Operational Assessment of our system during training scenarios and technology demonstrations at the 2010 Mock Prison Riot venue. The MPR is held annually on the grounds of the decommissioned West Virginia Penitentiary in Moundsville. The purpose of the Mock Prison Riot is to provide law enforcement and corrections (LEC) practitioners with opportunities for tactical training and exposure to the newest technologies available. The WVHTC Foundation's RespondComm Team, also a recipient of NIJ funding, provided a mobile elevated platform resulting in a "bird's eye" view of the system application area during the MPR. This approach leveraged existing efforts and kept resource and manpower costs down for similar NIJ projects. Our intelligent video system received high marks from LEC practitioners in the following areas: functionality, reliability, performance, and compatibility of the system in actual use during training scenarios and demonstrations. Housing units, recreational yards, and dining areas are high priority locations for system implementation. System attributes such as subject identification "numbering", facial recognition, and group type/gang identification also were highly valued by practitioners. Use as an intelligence gathering tool was applied to identify and determine types of contraband being transferred by individuals and groups and the methods used to conduct those transfers.

A major goal of our system is to perform continuous monitoring of locations, effectively detecting and possibly preventing crimes and minimizing personnel resources. Members of NIJ's Sensors and Surveillance Technology Working Group identified this capability as a high priority during their fall 2008 meeting.

12.1.1 Operational Parameters

For this assessment, operational parameters for GE Global Research Intelligent Video System were left to the discretion of the assessment team and on-site LEC subject matter experts, all of whom provided their feedback to the assessment Team.

A series of scenario scripts were used to guide the capture of specific interactions common in

a recreational corrections setting. These were conveyed to enacting officers who were given the option to enact these behaviors or any other behavior.

The following lists examples of events enacted during data collection. All events were enacted by correctional officers and role players. At no time were actual inmates involved in any part of this assessment.

1. Standard recreational yard behavior
2. Groups of prisoners arguing (pushing and shoving)
    - Assault on prisoner(s) by other prisoner(s)
3. Prisoner Suicide

The overall goal of the system is two-fold:

1. Detection: Detect an event (e.g., a fight). Detect presence of gangs. Detect group presence.
2. Prevention: Detect behaviors leading up to an event such that in the future, correctional staff has a chance to intervene before an event occurs.

The GE Global Research system looks for behaviors and patterns such as:

- Presence of multiple groups/gangs
- Agitation/aggression in yard
- Suspicious motion
    - Is one gang slowly approaching another gang?
    - Is a group of prisoners being Şflanked Ť in a suspicious way?
    - Is there a stand-off between prisoners?
    - Are two prisoners meeting (i.e., contraband exchange)?
    - Are prisoners charging or fleeing?
- Close-up imagery of prisoners

12.1.2 Results

During the assessment period, the GE Global Research Intelligent Video System was evaluated by practitioners from federal and state correctional institutions, regional and local jail systems, and law enforcement agencies. Feedback indicates the GE Global Research Intelligent Video System performed well on events for which it is designed. Refer to Section 12.1 for detailed evaluation.

Figures 53 to 57 illustrates a few examples of what the GE Global Research detected live during the enacted scenarios during the Mock Prison Riot.

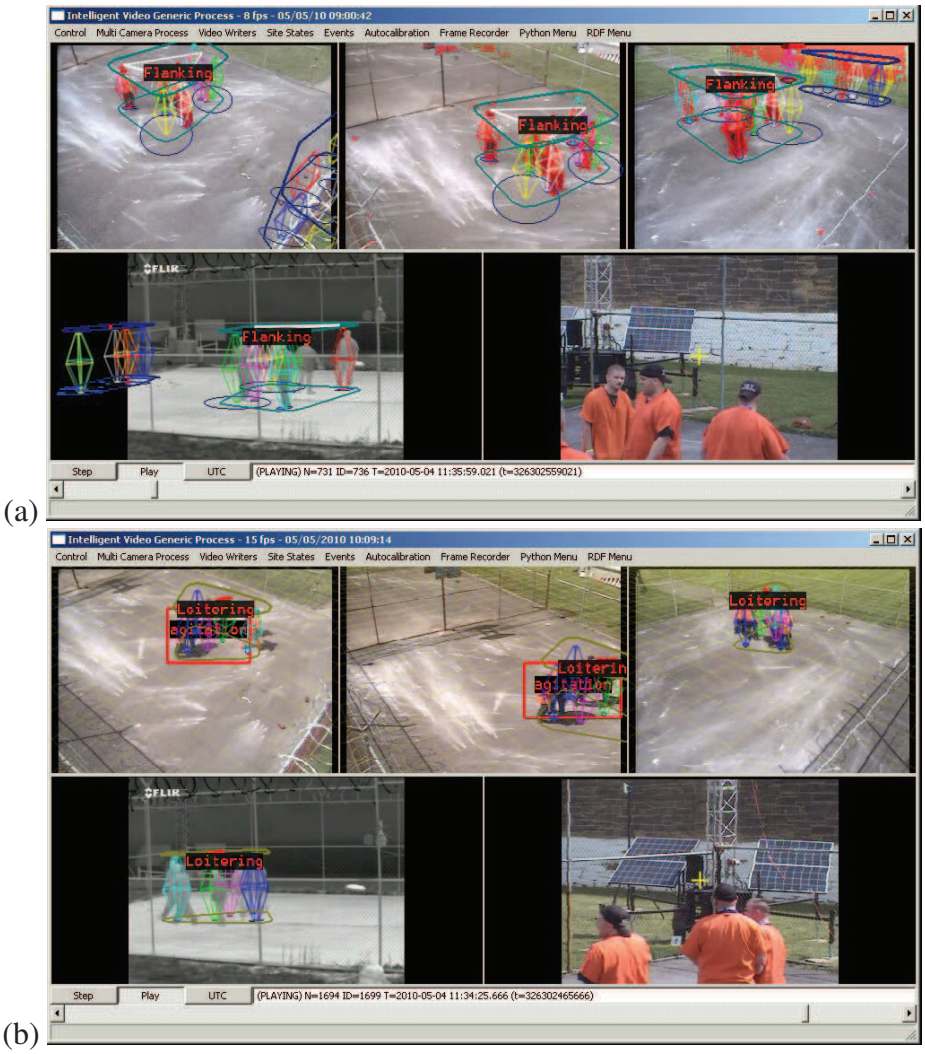


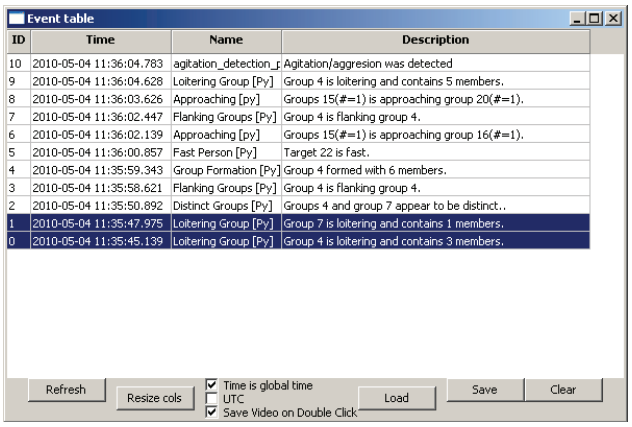
Figure 53: **Flanking event:** (a) Screen shot of our system detecting a flanking event just before a fight breaks out. (b) Detection of the actual fight. The screen short shows an enactment by Lake Erie correctional officers.

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

In the following, a detailed breakdown of a scenario that was analyzed and interpreted correctly by the system is provided:

Loitering detection: First, the system detected several loitering groups. The two images below show the event table as reported by the system (the loitering events are highlighted) and a screen-shot of the running system.

(a)



(b)

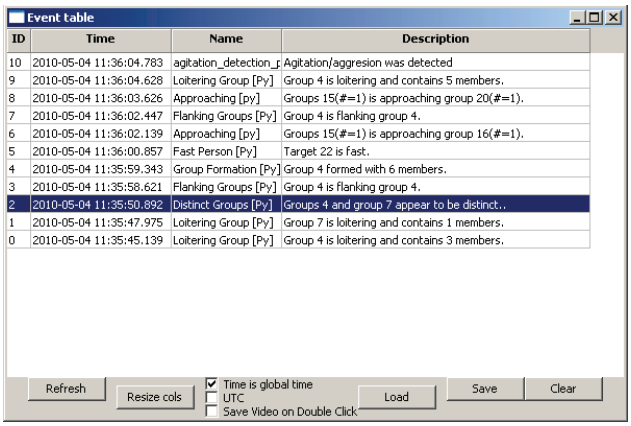
Figure 54: **Loitering event:** (b) Screen shot of our system detecting a loitering event: (a) The detected loitering event is stored in the event table. (b) The screen shot of the detected loitering event. The screen short shows an enactment by Lake Erie correctional officers.

Distinct group detection: The system established correctly the presence of multiple groups.

12.1.3 Issues / Performance Comments

Person Detection and Tracking. Below are the comments from Mike Lucey, Mock Prison Riot

(a)



(b)

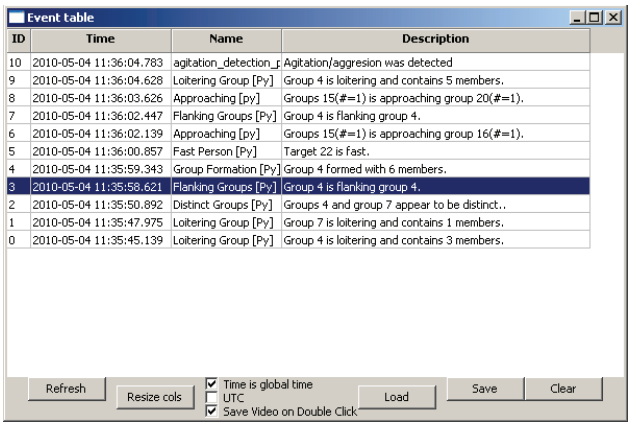
Figure 55: **Distinct groups detected:** Screen shot of our system detecting a distinct groups event: (a) Detected event stored in the event table. (b) The screen shot of the detection. The screen short shows an enactment by Lake Erie correctional officers.

Project Manager, from the MPR Operational Assessment Report:

"The person detection and tracking sub-component of the GE Global Research Intelligent Video system worked well, except under extreme conditions such as smoke and severe crowding. This is according to the expectations of the system. These obstacles are difficult, if not impossible, to overcome by a video-based system.

To investigate the possibility to overcome smoke, we have installed a thermal camera to firm individual's motion in the MPR 2010. In Figure 58 one can see that the smoke was essentially transparent in the thermal view.

(a)



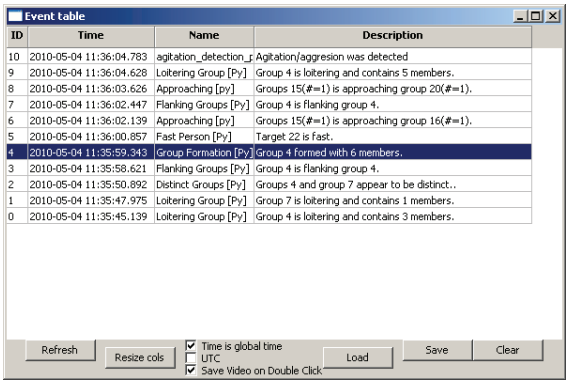
(b)

Figure 56: **Fight Prediction:** Prior to the fight, the system detected the flanking maneuver, which indicated that a fight is about to occur.

Event Recognition Performance. During live operation, the event recognition performance worked according to expectations. The system detected many actual behaviors during the scenarios as shown in the previous section above. Compared to 2009, the system detected many more relevant events live and in real-time. On several occasions, the system correctly detected and even predicted the presence of multiple gangs, loitering events, flanking approaches and fights. This system is optimized and improved continuously at GE Global Research. A more detailed quantitative performance assessment will be performed and reported by GE Global Research under the scope of their NIJ research program.

Active Camera Capture. The active PTZ camera, responsible for capturing high-fidelity facial shots of subjects, performed well. Compared to 2009 the alignment of the camera was greatly

(a)



(b)

Figure 57: **Group formation detected:** Detection of the formation of a larger group from the three separate pairs of (actors)inmates.

improved and weatherproof housing was used to mount the camera.

Other Scenarios. Teams suggested new scenarios that they would like the system to be able to detect in the future. There are no current functionalities in the system to detect such behaviors, but the guidance was useful for targeting future research efforts.

"Knife Sharpening": A person is bending down, sharpening a knife (shank) on the concrete, as illustrates in Figure 60.

"Suicide attempt": Detected as loitering, but a detector for sensing a person that is lying down or falling over would help in discriminating a loiterer from somebody that is committing suicide, as illustrated in Figure 61.

"Walkby": A person from a gang A does a "walkby" to a person from gang B, as illustrated in Figure 62. This behavior entails a sharp, almost invisible shoulder-to-shoulder hit while two people are passing. The current system is not able to detect subtle aggression such as this where one person bumps into another with his shoulder:

12.1.4 Feedback from Law Enforcement during MPR 2010

GE Global Research spend a lot of effort to show the running system to law enforcement and corrections group and to solicit feedback.

Stakeholder feedback was solicited around three commonly recognized benchmarks deemed

(a)

(b)

Figure 58: **Smoke:** (a) During almost all official MPR scenarios in 2009 (that were outside of the scope of the GE Global Research data collection), simulated smoke grenades were used. Smoke prevented the video analytics system from detecting people, which led to many false alarms and missed detections. System performance degraded severely. (b) Smoke are essentially transparent in the thermal camera view in the MPR 2010 collection.

critical to the evaluation of public safety technologies (DOJ Publication, 2002):

- **Functionality:** The degree to which the technology operated as described in response to user needs. Does it do what the scientists say it will do?
- **Reliability:** The degree to which the technology could be operated consistently under realistic field conditions. Do you feel it would perform reliably under real-world conditions?
- **Performance:** The degree to which the technology operated efficiently and timely relative to expected end user needs. Does it make the job easier?

Figure 59: **Condition that person tracking becomes impossible:** The ability to localize and track people under some conditions becomes impossible. There is a correctional officer in a heavily padded red man suit hidden in this picture.

Because most applications would require integration into already existing systems and networks, the issue of compatibility was added to the standard NIJ benchmark areas.

- **Compatibility:** The degree to which the technology can be added to the user's toolset without a negative impact on existing/traditional tools already in use. Will it cause adverse effects with other products or intended deployment areas?

Rating summary: On a scale of 1 (low) to 10 (high) in rating, the GE Intelligent Video System received high marks ("9" to "10") from evaluators in terms of performance, functionality, compatibility, and reliability of the system in actual use during training scenarios and technology demonstrations. The system also received high marks from LEC practitioners in the following areas: functionality, reliability, performance, and compatibility of the system in actual use during training scenarios and demonstrations. Housing units, recreational yards, and dining areas are high priority locations for system implementation. System attributes such as subject identification "numbering", facial recognition, and group type/gang identification also were highly valued by practitioners. use as an intelligence gathering tool was applied to identify and determine types of contraband being transferred by individuals and groups and the methods used to conduct those transfers. The most notable and immediate areas of system application were the entrance and corridors of the secure areas that complement existing access control systems within the correctional

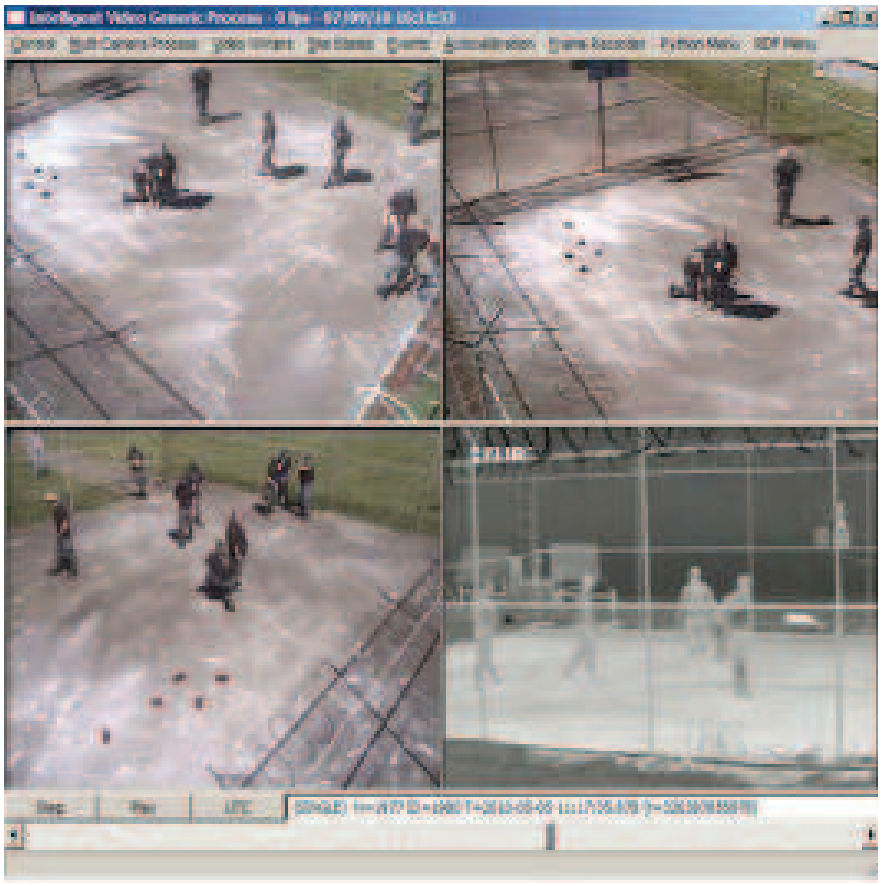


Figure 60: **Knife sharpening:** Partially shielded by another (actor) inmate, one (actor) inmate sharpens an improvised knife on the concrete ground.

facilities. Recreational, community gathering, and group interaction areas were noted as "ideal" environments offering the greatest likelihood of capturing illegal or suspicious activities. Almost all practitioners interviewed and identified the overwhelming role that actual event video plays in an effective and expedient prosecution process.

Guided by the four benchmarks, assessment team members were asked to evaluate how well the technology performed. Based on the totality of the responses, one of three subjective scorings was assigned to each area: Pass, Fail, or Reservation. "Pass" indicates the technology performed fully to the prescribed standard. "Fail" indicates the technology did not perform to the prescribed standard. "Reservation" indicates the technology performed essentially to the prescribed standard but still has operational issues to be resolved that would better satisfy user needs.

Scores for Operational Parameters: The following shows the systemŠs performance regard-

Figure 61: **Suicide attempt:** An enacted suicide attempt of a person sitting/slouching near the fence line. The person is visible in the top-left, bottom-left and bottom-right views.

ing several objectives. Note that PASS/FAIL ratings are viewed with respect to the current Technology Readiness Level of the technology.

Further developments are necessary to make the technology generalized and robust to all possible LEC environments.

A detailed analysis of system performance will be provided by the GE Global Research scientist working on the technology as part of his NIJ program reporting and dissemination process.

Further development of this technology is being closely monitored by the Sensors and Surveillance, Institutional Corrections, and Information-Led Policing Technology Working Groups.

12.2 System Deployment to a Local Police Department

We have tested the software and hardware configuration on a prototype system shown in Figure 64 that can be used for test deployments. This system includes a high-performance 12-core 2.6GHz

Figure 62: **Walkby:** An enacted "walkby" where one inmate (from one gang) hits the shoulder of another inmate (from another gang) while walking past.

HP Z800 worstation with 4GB RAM, equipped with a Matrox MorphisQxT framegrabber. Figures 65 and 66 show example screen shot of our behavior recognition GUI in performing surveillance.

We have successfully demonstrated our system on previous NIJ sponsored OLETC Mock Prison Roit events. Going forward, we initiated a new collaboration effort with the Schenectady City Police Department (SCPD) and the Schenectady County District Attorney's Office (DA) to deploy and validate our advanced behavior recognition system in a real-world Law-Enforcement site, the Public Surveillance Camera Project (PSCP). Both SCPD and DA agree that PSCP can be a testing site for GRC's analytic system, which will not only validate its performance in real-world setting, but also potentially provide value to SCPD's Law-Enforcement practices.

A three-stage system deployment and validation is planed:

1. SCPD selects a subset of its video archives that contain the recording of past events that have already been viewed and considered closed. SCPD transfers the videos to GRC. GRC

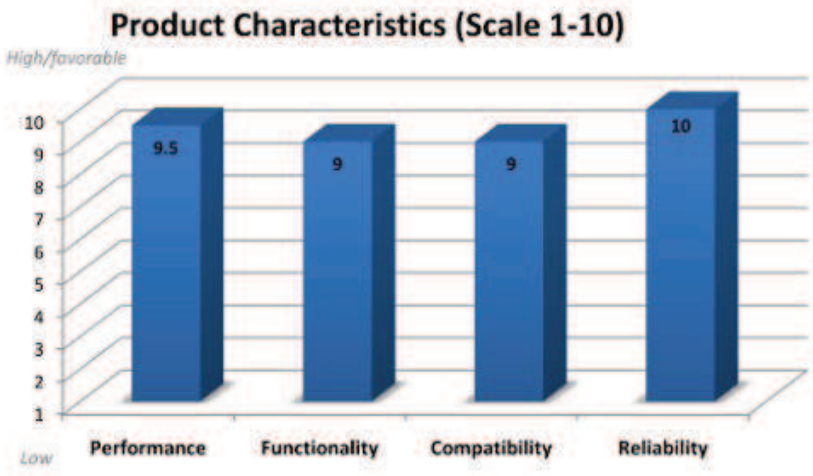


Figure 63: **Practitioner Rating.**

assesses system performance by processing these videos.

2. GRC deploys a workstation and software to SCPD at no charge to DA or SCPD. The PSCP feeds one or more videos directly to the GRC test station. GRC's system processes the video in real-time and records the analysis.

3. PSCP connects one additional video feed, as well as a PTZ control cable onto existing cameras, directly to the GRC workstation. GRC's system processes the video in real-time, records detected events of potential interest, and also automatically controls the PTZ camera to capture face/body images in higher resolution.

We have started preliminary data collection with SCPD. A sample video shot of some initial person detection and tracking results is shown in Figure 67. To focus on scenarios that are relevant to law enforcement, we will involve practitioners to identify scenarios and activities that the proposed system should be able to recognize automatically.

12.3 System Evaluation

Besides data collection and field testing at SCPD, we will also provide support for third party evaluation conducted by ManTech. Our systematic performance evaluation infrastructure is available

Table 6: **Operational assessment scores** from the participated practitioners in MPR 2010.

Presence of multiple groups/gangs	PASSED EXPECTATIONS
Agitation/aggression in yard	During event: PASSED EXPECTATIONS In contrast to 2009, the system now detected aggression live and in real-time.
Suspicious motion patterns	PASSED EXPECTATIONS
Is one "gang" slowly approaching another "gang"?	PASSED EXPECTATIONS
Is a group of prisoners being "flanked" in a suspicious way?	EXCEEDED EXPECTATIONS The system detected this pattern multiple times while running live on previously unseen data.
Is there a stand-off between prisoners?	PASSED EXPECTATIONS
Capture close-up imagery of prisoners.	EXCEEDED EXPECTATIONS
Operation during official public MPR scenarios (events outside of data collection scope of GE Global Research)	Without smoke: RESERVATIONS Some scenarios were extremely crowded and fast moving (20-30 SWAT officers + 10 inmates) still exceeding the capability of the video tracking system to track single individuals. However, groups of subjects are still tracked at a group level. With smoke: FAILED DUE TO SMOKE With heavy smoke, the system was not able to "see"subjects and the system lost the ability to visually track individuals. However a new thermal camera was utilized to assess the feasibility of tracking even in the presence of smoke, with promising results.

for continuously testing and evaluating the scenario recognition system. Systematic performance evaluation is important for intelligent video systems both in guiding the initial development as well as the final testing and optimization of algorithms. Extensive tools for groundtruthing video footage and comparing performance to groundtruthed data are available at the GE Global Research Center.

Training and testing data will be annotated manually at the event level. An interactive groundtruth tool will be used for this purpose, illustrated in Figure 68. For testing, events detected by the system will be compared against the ground truth and performance measures, such as detection rates, false alarm rates and precision/recall rates, will be evaluated and reported. Specifically, we will provide ManTech with the system test package including the software and a user manual for eval-

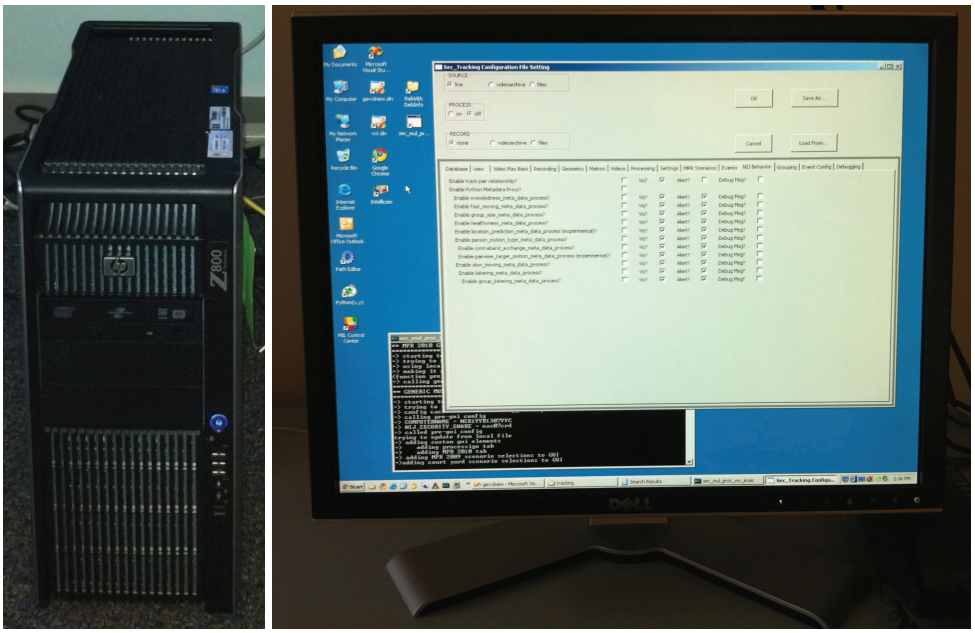


Figure 64: **Workstation for deployment and a GUI screen shot.**

uation purposes. The user manual will include detailed installation instructions for different test scenarios, including camera calibration procedures for (1) a single camera configuration or (2) multiple camera configuration. For both cases, a validation procedure will be provided to verify the calibration results. Sample testing sequences will be included in the test package and the corresponding expected results and troubleshooting instructions will be provided in the manual. For example, screen shots similar to Figure 69 that shows the expected event labeling and detection results on a test sequence will be provided. We plan to deliver the test package to ManTech between December 2011 and March 2012.

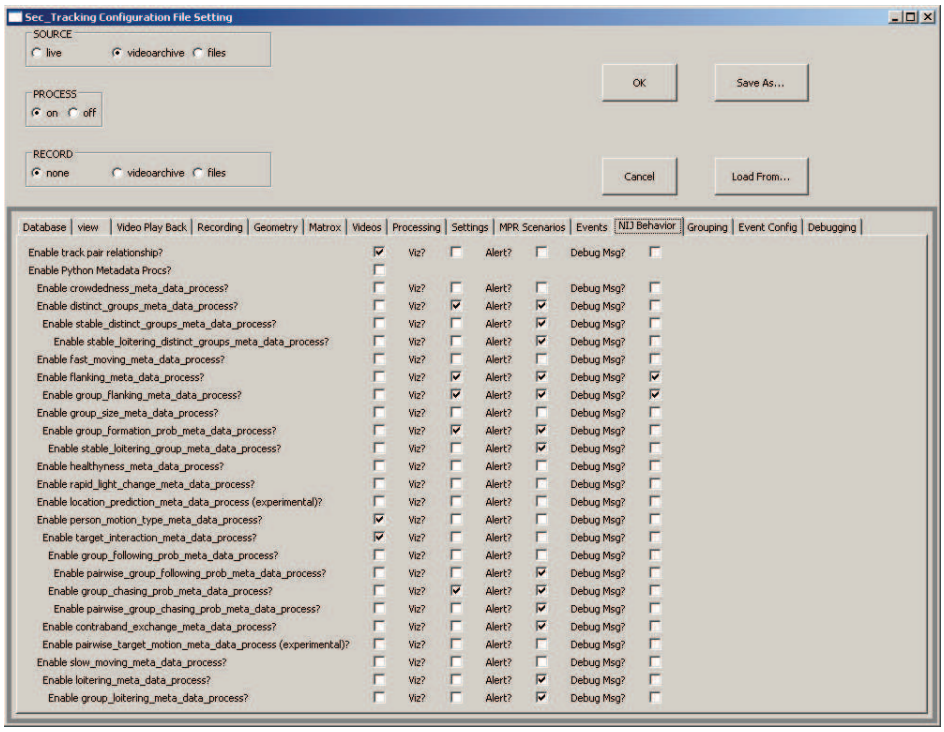


Figure 65: **System operation GUI for video acquisition and algorithm settings.**

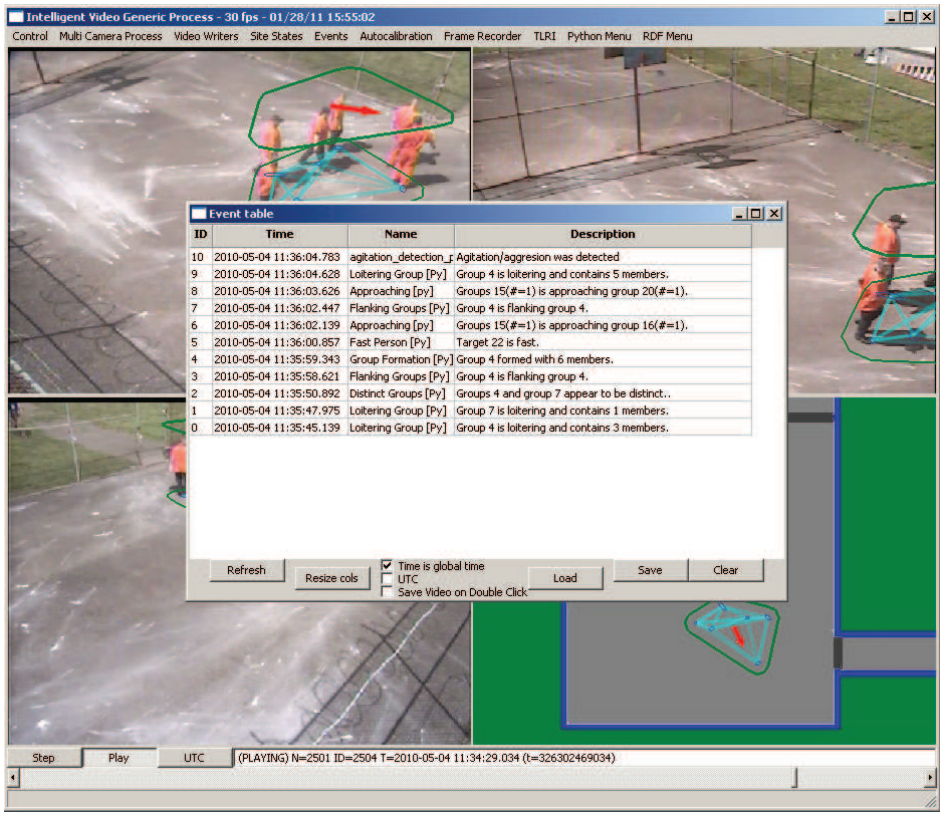


Figure 66: **Multi-view display of individuals under tracking and monitoring, where event alerts are shown in a table in real time.**

Figure 67: **Real-world Law Enforcement Data Collection.** A sample video frame from the Public Surveillance Camera Project (PSCP) operated by the Schenectady City Police Department (SCPD).

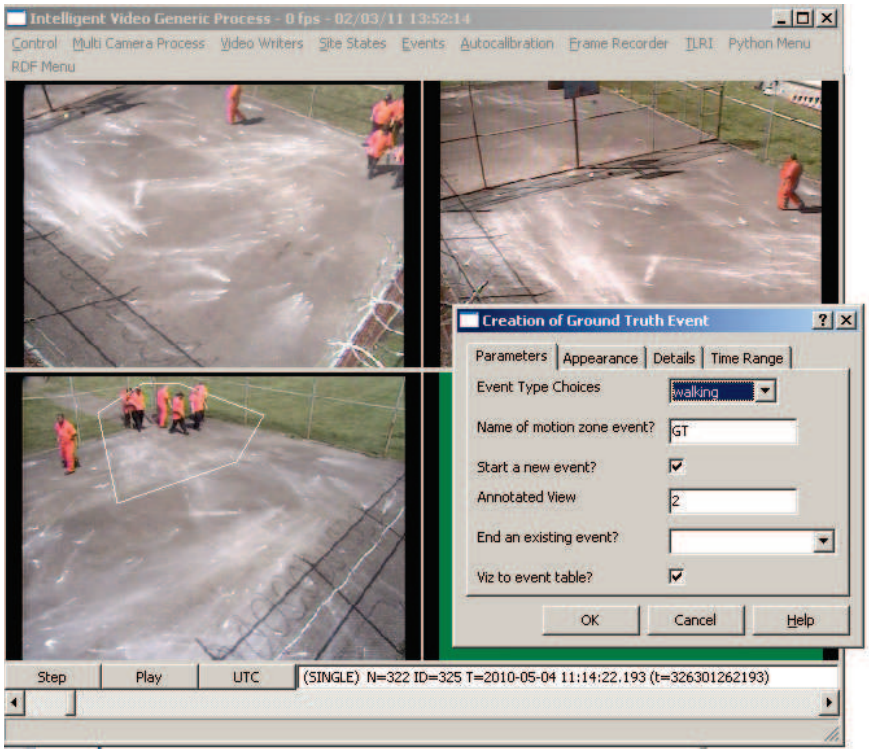


Figure 68: **Groundtruthing for System Evaluation.** A sample screen shot showing the manual event labeling interface.

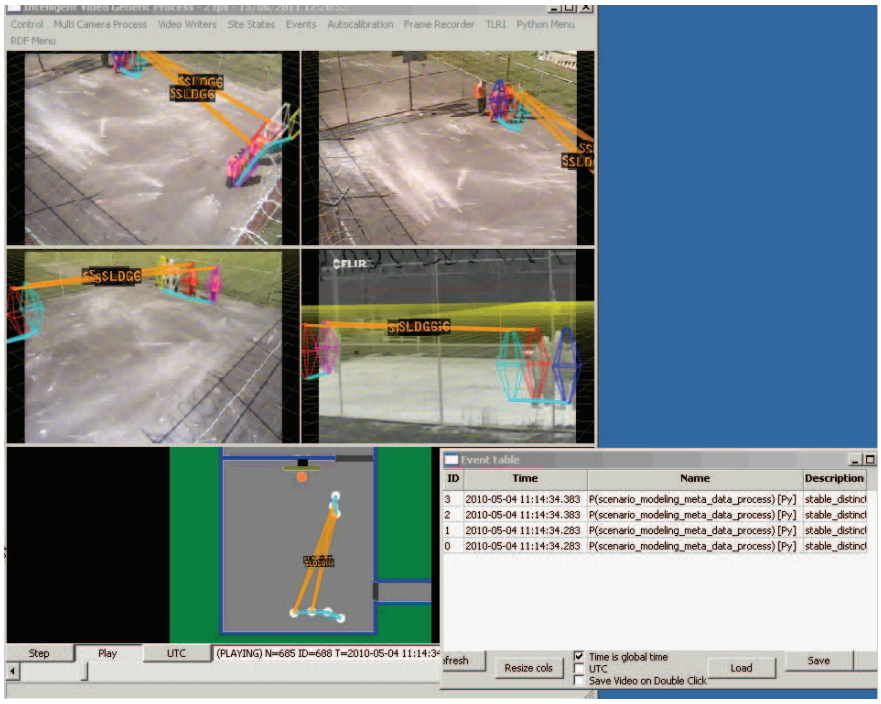


Figure 69: **Sample Event Detection GUI Front End.** A sample screen shot showing the expected event detection results on a test sequence. All triggered events are stored in an event table, which allow convenient retrieval of the underlie video clips containing the detected events.

A Public Dissemination

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

[1] Ming-Ching Chang, Nils Krahnstoever, Sernam Lim, and Ting Yu, "Group Level Activity Recognition in Crowded Environments across Multiple Cameras", In Proc. Seventh IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), Workshop on Activity Monitoring by Multi-Camera Surveillance Systems (AMMCSS), Boston, MA, pp. 56–63, Aug.-Sep., 2010.

[2] Karthik Sankaranarayana, Ming-Ching Chang, and Nils Krahnstoever, "Tracking Gaze Direction from Far-Field Surveillance Cameras", In Proc. IEEE Workshop on Applications of Computer Vision and Applications (WACV), Kona, Hawaii, pp. 519–526, January, 2011.

[3] Nils Krahnstoever, Ming-Ching Chang, and Weina Ge, "Gaze and Body Pose Estimation from a Distance", In Proc. Seventh IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), Klagenfurt, Austria, August, 2011 (Best Paper (Runner Up) Award).

[4] Ming-Ching Chang, Nils Krahnstoever, and Weina Ge, "Probabilistic Group-Level Motion Analysis and Scenario Recognition", In Proc. IEEE 13th International Conference on Computer Vision (ICCV), Barcelona, Spain, Nov., 2011.

The following provisional patent related to part of efforts on gaze tracking is in progress of filing:

Nils Krahnstoever, Peter Tu, Ming-Ching Chang, Weina Ge, "Person Tracking and Interactive Advertising", Provisional Filing, Application Serial No. 13/221,896.

As part of the public dissemination, the work covered in this grant has been featured on the front page of the New York Times (Figure 70).

Figure 70: Article that Appeared in the New York Times.

B Reviews and Meetings

B.1 Mock Prison Riot 2010

The main Mock Prison Riot Event took place in May, 2010. 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.2 Program Review 2010

On August 19th, 2010, Francis Scott, Sensors and Surveillance Portfolio Manager at the National Institute of Justice, visited the GE Global Research Center (GRC) for a program review. Nils Krahnstoever, Ming-Ching Chang, and Peter Tu presented an overview of the current program together with a live demo on group-level event detection at the GE GRC Courtyard, where four fixed surveillance cameras and four Pan-Tilt-Zoom (PTZ) cameras are equipped. Scenarios including loitering, slow moving individuals, fast moving individuals, and contrahand handoff events are all successfully demonstrated in live action.

B.3 IEEE Conference on Advanced Video and Signal-Based Surveillance (AVSS) 2010

Nils Krahnstoever and Ming-Ching Chang attended the IEEE Conference on Advanced Video and Signal-Based Surveillance (AVSS) in Boston, MA on August 29 to Sep. 1, 2010 to present the paper "Group Level Activity Recognition in Crowded Environments across Multiple Cameras" in the Workshop on Activity Monitoring by Multi-Camera Surveillance Systems (AMMCSS).

B.4 IEEE Worpshop on Applications of Computer Vision (WACV) 2011

Ming-Ching Chang attended the IEEE Worpshop on Applications of Computer Vision (WACV) in Kona, Hawaii on Jan. 2011 to present the paper "Tracking Gaze Direction from Far-Field

Surveillance Cameras".

B.5 Program Review 2011

On June 20, 2011, Xiaoming Liu had a program review with Francis Scott and Lars Ericson from ManTech. Xiaoming Liu presented the progress of the NIJ program in the past year, in particular on the event recognition, event explanation and scenario modeling GUI.

B.6 NIJ Conference 2011

Xiaoming Liu (PI) presented a poster of the program at the NIJ conference on 06/20-22, 2011. The poster was well received and we established valuable contacts to law-enforcement practitioners interested in video surveillance technology.

B.7 2011 Technologies for Critical Incident Preparedness Expo (TCIP)

Xiaoming Liu visited the TCIP Expo on 08/29-30,2011 and presented current results and findings of the NIJ program to the law enforcement community. The exhibition in the form of a booth was very successful and visited by a large number of customers. The video presentation and exhibit material was received favorably, and several useful connections with the community have been established.

C Group Level Activity Recognition

The following is a reprint of a paper that will be presented and published at the Workshop on Activity Monitoring by Multi-Camera Surveillance Systems in conjunction with AVSS 2010 in Boston. The precise title of the paper is: Ming-Ching Chang, Nils Krahnstoever, Sernam Lim, Ting Yu, "Group Level Activity Recognition in Crowded Environments across Multiple Cameras", In Proc. Workshop on Activity Monitoring by Multi-Camera Surveillance Systems (AMMCSS), Boston, August, 2010.

Abstract

Environments such as schools, public parks and prisons and others that contain a large number of people are typically characterized by frequent and complex social interactions. In order to identify activities and behaviors in such environments, it is necessary to understand the interactions that take place at a group level. To this end, this paper addresses the problem of detecting and predicting suspicious and in particular aggressive behaviors between groups of individuals such as gangs in prison yards. The work builds on a mature multi-camera multi-target person tracking system that operates in real-time and has the ability to handle crowded conditions. We consider two approaches for grouping individuals: *(i) agglomerative clustering* favored by the computer vision community, as well as *(ii) decisive clustering* based on the concept of modularity, which is favored by the social network analysis community. We show the utility of such grouping analysis towards the detection of group activities of interest. The presented algorithm is integrated with a system operating in real-time to successfully detect highly realistic aggressive behaviors enacted by correctional officers in a simulated prison environment. We present results from these enactments that demonstrate the efficacy of our approach.

C.1 Introduction

The capability to automatically detect suspicious, disorderly or criminal activities from video sequences is highly desirable in domains such as prisons, schools, public places, sport venues and

other public gatherings. Even more appealing than the detection is the early *prediction* of events that allows action to be taken before an event unfolds or escalates. We are particularly interested in domains and activities relevant to law-enforcement, and are addressing here the problem of detecting behaviors in environments where many close interactions occur at a group (or rather gang) level. Hence we seek to address the problem of detecting and reasoning about the spatio-temporal evolution of group structures so as to understand group-level activities of the crowd.

In computer vision, typical grouping strategies [30, 31, 32] rely on a bottom-up, *agglomerative* clustering [18, Ch.10.9] of individuals in order to find the group structure. Such grouping schemes could be *hierarchical* based on previously established clusters, using some distance metric reflecting the spatial-temporal features of the tracked targets. Hierarchical clustering approaches require a threshold (stopping criterion) to determine the final grouping. This is appropriate in environments where observed person-to-person distances follow standard social norms (*i.e.*, *proxemics* [2]) but not in environments where rapid changes in interaction distances occur [33, 34]. In order to compensate for the latter, we consider a second approach based on an eigen analysis of the graph adjacency matrix, which is motivated by its success in the social network analysis community. We propose a method based on the top-down concept of the graph *modularity* measure [34], which maximizes the difference between the connections within a group of individuals and the expected number (and strengths) of such connections. The grouping is performed by dividing the graph along connections that are not necessarily weak, but rather *weaker than expected*. This is crucial in achieving an *adaptive* grouping to segment groups across a variety of configurations, which is essential in this work.

In this paper, we present algorithms for recognizing several group-level activities that are of particular interest to the law-enforcement community. These algorithms range from recognizing low-level activities such as group formation, group dispersion, group loitering, to more advance activities such as group flanking and aggression/agitation. The presented algorithms are integrated in a comprehensive real-time surveillance system that performs multi-camera, multi-target tracking in challenging environments. The overall framework has been evaluated and tested live in an

abandoned former prison with professional correctional officers enacting typical inmate behaviors. The system was able to successfully detect a variety of group level activities, even successfully *predicting* the occurrence of (simulated) aggressive behaviors (gang on gang fights) before the actual event unfolded.

The rest of this paper is organized as follows. We discuss related work in Section C.2. Section C.3 briefly outlines our tracking system as well as the domain we are addressing in this paper. Section C.4 describes the two strategies (bottom-up, top-down) for determining group structures. Section C.5 describes the utilization of these group structures for recognizing group activities. In Section C.6, we report test results from our system detecting in real-time group events in simulated law-enforcement environments. Section C.7 concludes this paper.

C.2 Related Work

Fundamental to the success of any algorithms for recognizing group activities is the ability to track individuals (or group of individuals) under crowded conditions. There are numerous works that address the tracking problem both at the individual [35, 36] and group level [30]. In fact, whether or not a tracked blob belongs to an individual or a group could be ambiguous due to heavy occlusion. To this end, Grimson *et al.* [37] detect and track multiple objects as moving blobs and disambiguate fragmentation/over-segmentation by building an inference graph; from which they reason about the entire tracks of the objects based on spatial connectedness and motion coherence.

Given a set of detected tracks of individuals, one can group these tracks into cohesive entities. Ge *et al.* [31] identify small group structure of a crowd in a bottom-up fashion by iteratively merging sub-groups with the strongest inter-group closeness, utilizing a measure based on the symmetric Hausdorff distance. The clustering is hierarchical and is similar to the construction of a minimum-spanning tree (MST) from the individuals. The use of the Hausdorff distance requires continuous recomputation of group-to-group distance measures, which can be an expensive operation for large multi-camera surveillance sites.

As opposed to grouping individual tracks, there are also algorithms that identify grouping with-

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

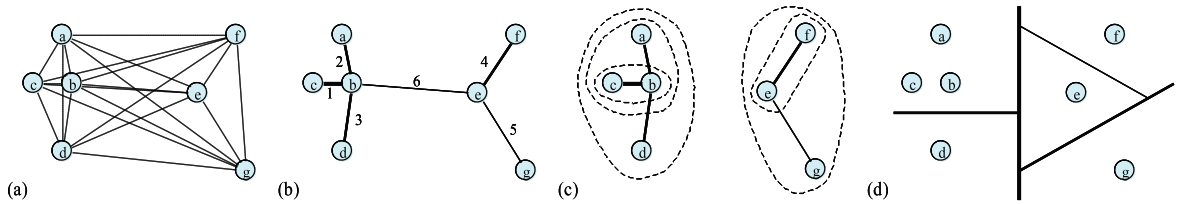


Figure 71: **Hierarchical *agglomerative* and *divisive* clustering.** (a) Determining the grouping of several individuals (a to g) is combinational in nature; here a complete graph is depicted. (b) Agglomerative clustering following Kruskal's algorithm in constructing the MST of the individuals. Edge weight reflects the distance between individuals. A "hierarchy" of groups corresponding to the disjoint forest sets in the MST is depicted in (c). (d) Divisive clustering formulates the grouping as a recursive cutting problem, where at each step the optimal cut between two subgroups is determined.

out necessarily identifying individuals in the groups. Lau *et al.* [30] hypothesize over both the partition of tracks into groups and the association of detections into tracks, and pose the group modeling problem as a recursive multi-hypothesis model selection problem. Groups are formed using *single linkage clustering*, which also is a variant of the MST algorithm. The assignment of observed cluster to a group is estimated using the minimum average Hausdorff distance. The merging of two groups is justified using a Mahalanobis distance between closest contour points. Hard thresholds on the blob size are used to identify whether a blob is a person or a group of people.

Robust detection and tracking of groups allows for the recognition of group activities and behaviors. Saxena *et al.* [32] model crowd events by defining case-specific scenarios and detect abnormalities such as falling, fighting, and emergence of new crowd flow. The event is trigger by imposing a hard threshold on several measures including crowd density, principal directions, number of individual motion vectors in a crowd.

C.3 System and Site Description

In this section we will provide a brief outline of our tracking system as well as the type of environment we are addressing in this paper.

C.3.1 Video Tracking System

A key factor for successful group analysis and subsequent recognition of group activities is the efficacy of the underlying video tracking system. The tracking system must perform reasonably well even under heavy occlusions, since groups often form in crowded conditions.

The tracking system that we utilize [38, 39, 36, 40] comprises of multiple calibrated static cameras tracking cooperatively in a synchronized fashion. For each view, based on the calibration, the image dimensions and positions of a given person at all possible 3D locations in the scene are estimated. These image locations are precomputed, and foreground pixels detected during online tracking are used to vote for these precomputed image locations to form a set of (foreground) detections [38]. This effectively leverages the calibration information to significantly reduce false positives arising from occlusions and crowdedness.

The set of detections for each view are then projected onto the ground plane in 3D in order to further disambiguate any confusion due to occlusions and crowdedness. These projections are consumed by a centralized tracking system that either (1) associates detections with existing tracks based on spatial proximity or (2) initiates new tracks. The states of tracks are estimated by a standard Kalman filter, performed in the world reference ground plane. The system is designed to maintain tracks across camera boundaries in order to perform site-wide tracking. Through calibrated camera views, all tracking information can be visualized in a top-down view of the whole surveillance site , which is particularly useful for security operations.

C.3.2 Domain

The system presented here is aimed at detecting suspicious and disorderly behaviors. For data collection and testing purposes the system was deployed in an abandoned prison yard in West Virginia, USA. Several correction officers volunteered to enact domain relevant behaviors such as agitated arguments, fights, contraband exchange and many others. As many activities of interest for correctional settings are related to gang activities, many of the enacted scenarios simulated the presence of multiple gangs. The test system utilized a total of 4 standard CCTV cameras with three

cameras used for tracking, one camera for automatic PTZ targeting (which will not be discussed in this work). An optional fifth camera was used for thermal imaging. The system performed live processing and reported events of interest in real-time to the operator.

C.4 Group Analysis

Given a set of tracked individuals, the first step towards group activity recognition is to cluster individuals into cohesive groups. This step plays a critical role in accurately detecting group-level events and recognizing group activities later on. Cluster analysis is well-studied in pattern classification and serves as a common technique in many fields. We focus on *hierarchical* clustering [18, Ch.10.9], which is simple in concept where successive clusters are found using previously established clusters. The clustering efficacy can be adaptively refined in an recursive fashion.

Hierarchical clustering can be divided into two main categories: agglomerative and divisive [18], Figure 71. *Agglomerative* clustering operates bottom-up, starting with each individual as a separate cluster and merging them into larger clusters. *Divisive* clustering operates top-down, beginning with the whole set and dividing it into smaller clusters. We investigate both approaches in the context of monitoring the (social) group structures of tracked individuals as follows.

A major component in group clustering is how the distance measure between individuals is defined. In agglomerative clustering, the distance function is often a fixed measure between two individuals such as the commonly used Euclidean (2-norm), Manhattan (1-norm), or maximum norm metric. It can be a variable measure depending on the current clustering configuration *e.g.* Hausdorff or Mahalanobis distance. In divisive clustering, finding the best division is often casted as finding the best cut in a graph network, where the distance is treated as edge weights and graph-theoretic methods can be directly applied.

C.4.1 Hierarchical Agglomerative Clustering

The agglomerative nature in clustering suggests a simple and intuitive way to form groups from individuals. We consider here the spatial-temporal dissimilarity between tracks of individuals as

distance measure. First, a pair of elements with minimum distance are grouped together; then the second closest pair (which could be the newly formed group or a third) is merged; this process is repeated until a stopping criteria (distance threshold θ_m) is reached. Such clustering is bottom-up, local, greedy, and hierarchical, and is essentially constructing a minimum spanning tree (MST) of the individuals based on Kruskal's algorithm [13, Ch.24]. As Figure 71(b-c) illustrates, the intermediate groups and the hierarchy of subgroups correspond exactly to the disjoint forest sets generated by Kruskal's algorithm. The MST can be computed efficiently in $O(E \ln V)$, where V is the number of individuals and $E = O(V^2)$ is the number of edges of a complete graph of V nodes. where the edges of the complete graph is the upper diagonal matrix of M.

Result of the MST clustering, *i.e.,* the disjoint forest set provides a naive hierarchical group representation, Figure 71(c). In this representation, an individual can be assigned to many groups in the hierarchy. Group attributes such as geometric center, size (variance), and number of individuals can be computed at different grouping scale by tuning the threshold θ_m.

A major limitation of agglomerative clustering is that weaker connectivity is never considered in the clustering process. Figure 72 depicts an example, where five individuals a, ... e in a ring are clustered following the MST edges with weights 1, 2, 3, 4, respectively. Edge $\overline{ae}$, the closest path between a and e is never considered in the MST grouping process.

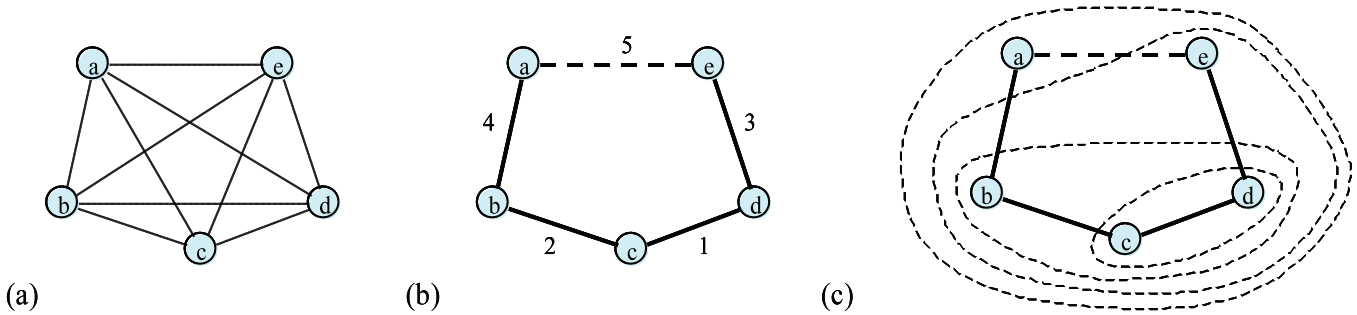


Figure 72: **Limitation of agglomerative clustering following the MST scheme.** (a) complete graph of five individuals; (b,c) result of the MST clustering is not optimal, since edge $\overline{ae}$, the closest path between a and e, is not part of the MST.

C.4.2 Hierarchical Divisive Clustering Using Modularity Cut

An alternative approach towards group analysis is to go top-down, to recursively divide the individuals into subgroups in a way that individuals with strong connections are placed in the same group. From a graph-theoretic perspective, the problem is to divide the complete graph containing all individuals as nodes into subgraphs in a way that maximizes within-group connections and minimizes between-group connections. A closer look at the problem should reveal the applicability of several well-known spectral clustering techniques [41, 42, 43, 44]. 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* [43].

In this paper, we apply a developed technique in social network analysis [40] to the problem of group structure formulation. We propose that instead of using cut size as the criterion, we adopt an approach originally proposed by Newman [45, 34] in the domain of social network study. Newman 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* [45, 34, 46], which expresses the difference between the actual and expected connections of individuals within each social group. Inherently, since individuals group together because of common social characteristics, such a modularity measure appropriately captures group analysis from a social perspective.

Consequently, two individuals, i and j, are strongly connected only if their connection B_{ij} is stronger than what would be *expected* between any pair of individuals, that is,

$$B_{ij} = A_{ij} - \frac{k_i k_j}{2m}, \tag{70}$$

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 complete 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. 70, the *modularity measure*

Q, can be derived as

$$Q = \frac{1}{2m} \sum_{\substack{i,j \in \\ \text{same group}}} B_{ij} = \frac{1}{4m} \mathbf{s}^T \mathbf{B} \mathbf{s}, \tag{71}$$

where $\mathbf{s}$ is a labeling vector whose element s_i corresponds to an individual (node) in the complete 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 $\mathbf{B}$.

Determining $\mathbf{s}$ that maximizes Q is shown to be NP-hard [34]. However, one can closely approximate the optimal solution by deriving the eigen decomposition $\mathbf{B} = \sum_i \beta_i \mathbf{u}_i \mathbf{u}_i{}^T$ with eigenvalues β_i and eigenvectors $\mathbf{u}_i$, and assigning s_i to $+1$ if the corresponding element in the maximum eigenvector is positive, and -1 otherwise. This has been shown in [34] to work well in practice.

The strategy for dividing a group into two subgraphs can be applied recursively to divide a group into an arbitrary number of hierarchical subgroups. To do so, we first define a $n \times c$ binary matrix $\mathbf{S}$, where n is the number of nodes in the complete graph and c is the number of groups. We begin with $c = 1$, i.e., there is only one group (the entire graph). As c increases, we recursively divide the graph into multiple groups. The $(i, j)^{th}$ element of $\mathbf{S}$ is 1 if node i belongs to j, and 0 otherwise. The modularity can be equivalently measured as

$$Q = \mathbf{Tr}(\mathbf{S}^T \mathbf{B} \mathbf{S}), \tag{72}$$

where $\mathbf{Tr}$ represents the trace operator. Based on Eq. 72, the strategy for dividing into multiple groups is as follow. Each time we obtain a new group, 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

$$\Delta Q = Q' - Q_{max}, \tag{73}$$

Event category	Events
Group detection	Formation, Dispersion, Distinct Groups
Motion pattern	Loitering, Fast Moving, Approaching, Following
Behavior event	Flanking, Agitation, Aggression

Table 7: List of detectable group-level events.

such that if $\Delta Q \leq 0$, the new group is "discarded" and the stopping criterion met.

In this top-down, cut-based group clustering scheme, group attributes such as center, size, members of a group are explicit. Given arbitrary configuration of individuals under tracking, the modularity cut stops when no better cut can be found. There is no parameter required in applying the modularity cut, which is very suitable to monitor rapid changes of group-level and individual interactions in this work.

The grouping of targets is performed on a per-frame basis. We explicitly keep track and maintain the temporal history of all groups and their members. Our system is thus capable of reasoning over the history of 'split-and-merge' of groups over time to reinforce the putative group behaviors. We will compare the performance of both agglomerative and divisive clustering for group-level event recognition in Section C.6.

C.5 Group Activity Recognition

As described in Section C.3.2, the aim of this paper is to detect, recognize and even predict group behaviors. Based on data observations and feedback from domain experts, the events and activities addressed by our system include low-level ones such as (i) group formation, (ii) group dispersion, and (iii) loitering. We also propose to detect semantically more advanced activities including (iv) approaching, (v) flanking, and (vi) aggression/agitation within groups, see Table 7. In the following we will describe a subset of the above events in more detail.

The event of *loitering* is detected by analyzing the standard deviation of the group location across a time window ($T_{\text{loi}} = 10$ sec). The group location for a given time is given by average ground location of all members in the group. If both components of the standard deviations are

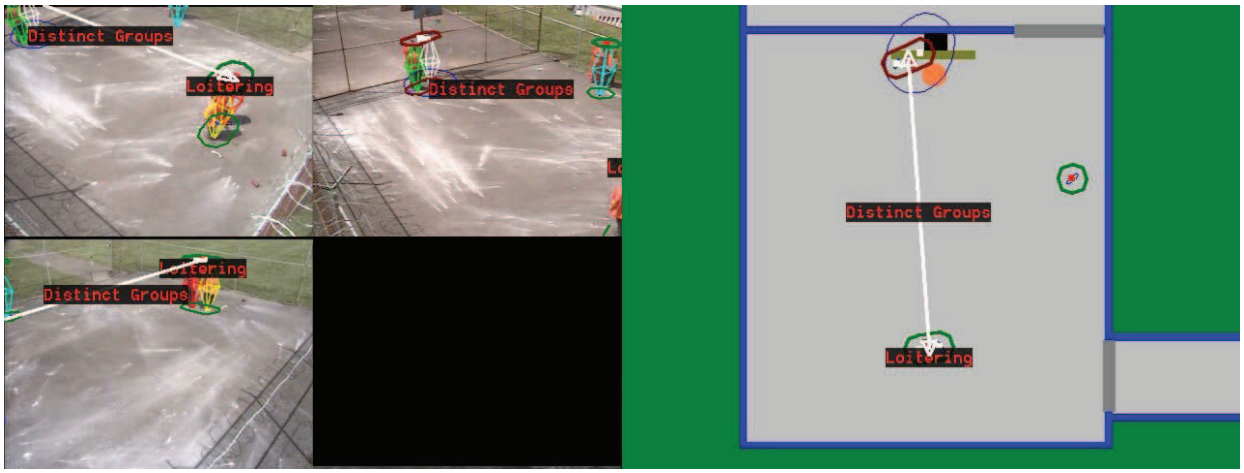


Figure 73: Detection of Distinct and Loitering Groups. Distinct groups are detected. (Left) Three views from synchronized cameras (one view masked out for privacy reasons). (Right) A fused planar view from the top, where distinct groups are detected and a loitering group is highlighted. Note the advantage of using a multi-view camera tracking system. Even though one of the distinct groups is outside each of the three camera views, the system still successfully detect such events. The shown activities have been enacted by law enforcement and corrections personnel.

below a given threshold ($\tau_\sigma = 0.5$ m), a loitering event is generated. Based on *loitering* a second related event is defined, namely the *distinct groups* detection. Distinct groups are defined as pairs of loitering groups that maintain a stable membership set for a period of time ($T_{\text{dis}} = 2.0$ sec) and are within a certain reach of each other ($d_{\text{dis}} = 10.0$ m). The presence of multiple distinct groups indicates an increase in overall intra-group cohesion, which in turn raises the possibility of inter-group conflict. See Figure 73.

The event *group formation* is detected by counting the ancestors of a group within a certain time window ($T_{\text{gf}} = 3$ sec). A group has to form from at least three ancestors. And the ancestor groups must have existed for a minimal amount of time ($T_{\text{exist}} = 2$ sec). No further constraints need to be imposed on spatio-temporal relationships. The event of *group dispersion* is similar.

The event of *group flanking*, or flanking maneuver (groups surrounding another group prior to an attack) is aimed at detecting a certain spatio-temporal configurations that is exhibited by groups before they engage in aggressive behaviors (see Figure 74). Data seems to indicate that an aggressive and dominating (in terms of strength and numbers) group tends to "surround" the victim group or individual or at least spatially spread out before the event. Flanking is detected as

follows.

We denote with $G = \{G_i, i = 0, \ldots, N_g - 1\}$ the set of all groups and with $G_i = \{T_{ij}, j = 0, \ldots, N_t^i\}$ the set of all tracked individuals in group G_i. We denote with $T = \{T_{ij}\}$ the set of all individuals. Furthermore, let $\mathbf{X_i}$ and $\mathbf{X_{ij}}$ be the locations of G_i and T_{ij} respectively. We now consider all triplets (G_i, T_{jk}, T_{lm}) and consider group G_i to be flanked if the following conditions are met:

- T_{jk} and T_{lm} are either in same group (i.e., $j = l$) of size $N_f := N_j$ or are direct descendants of the same group (of size N_f) and this group is not an direct relative of G_i.
- Group size N_f is at least 2.
- The angle between $\mathbf{X_{ijk}} = \mathbf{X_{jk}} - \mathbf{X_i}$ and $\mathbf{X_{ilm}} = \mathbf{X_{lm}} - \mathbf{X_i}$ exceeds a minimal angle θ_{fl}.
- The distance $d_{jklm} = ||\mathbf{X_{jk}} - \mathbf{X_{lm}}||$ between T_{jk} and T_{lm} must exceeds $d_{ijklm} = (||\mathbf{X_{ijk}}|| + ||\mathbf{X_{ilm}}||)/\mathbf{2}$.

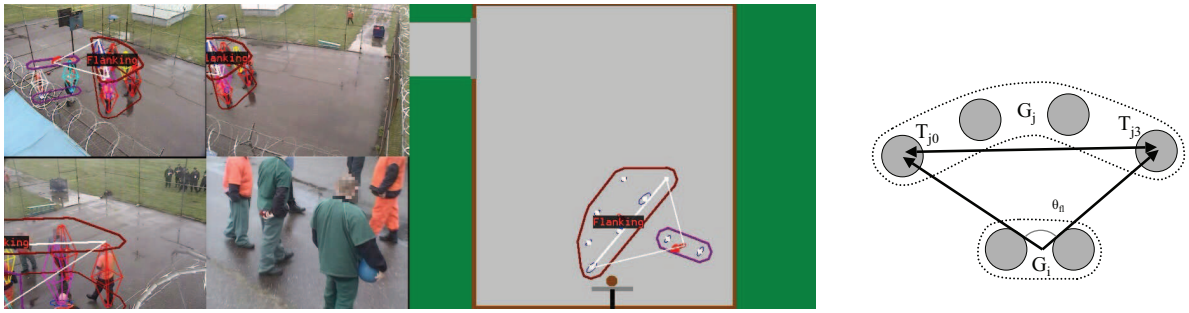


Figure 74: Flanking Detection. An event where one group is surrounding another group. Left: Example of a real-life flanking event in three synchronized views and one zoom-in view. Middle: A planar top view. Right: Schematic model of the flanking event. The shown activities have been enacted by law enforcement and corrections personnel.

The detection of *aggression/agitation* is different than the group-level events introduced so far, in that it does not operate on top of the tracking system but rather operates independently to detect image regions that contain agitated or aggressive behaviors. Only the observation of the spatio-temporal movements of individuals and groups is not sufficient to determine if agitated behavior is being exhibited. Rather, we perform sparse feature tracking using in the *foreground* of the scene

and classify aggression/non-aggression based on features extracted from these tracks. We utilize the FAST feature detector developed by Rosten and Drummond [47] for low-level point detection. To obtain trajectories for the detected points, we developed a data association-based point feature tracker that utilizes a fast greedy approach to perform detection to track association. After the tracking step is done, every trajectory is analyzed with regards to a range of motion attributes and the attributes are accumulated in local "decision blocks" (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 number of example sequences to obtain an optimal classification. Figure 75 shows a set of examples where aggression was detected in scenarios that were enacted by correctional officers. It should be pointed out that unlike the tracking system, the agitation detection operates on each camera view independently. See figures 75 and 76 for examples of successfully detected events in the prison dataset presented in this work as well as the BEHAVE dataset [48].

C.6 Experiments and Results

We are presenting first experimental results on detecting behaviors in challenging multi-camera surveillance environments with a focus of detecting (i) the presence of gangs, (ii) the prediction of a possible fight, and finally (iii) the detection of agitated motion patterns that are indicative of a fight.

Figure 77 shows a sequence lasting about 1 minute where two smaller groups (a gang approaching from different angles) is approaching and then attacking two individuals. The correct sequence of events was recognized. The event evolves very quickly and there is only a gap of about 1.5 seconds between the detection of the flanking maneuver and the onset of the fight. Figure 78 shows a similar scenario where a single group is approaching and then attacking another. The system again detected the correct sequence of events (only flanking and fighting shown) and predicts the onset of a fight three seconds before the actual fight breaks out. It should be noted that the same result was obtained with the system capturing *and processing* the scenario live on-site during the

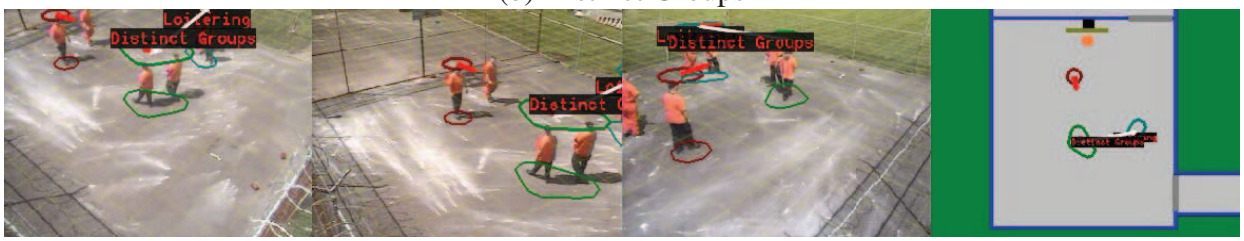
All Mock Prison Riot data was *enacted* by law enforcement and corrections practitioners.

Figure 75: Aggression Detection. Examples of regions that were classified to contain agitated or aggressive behavior. Approximately 2 seconds elapse between the frames on the left and the frames on the right. The shown activities have been enacted by law enforcement and corrections personnel.

Figure 76: Aggression Detection. Example aggression events detected in the BEHAVE dataset [48].

(a) Loitering

(b) Distinct Groups



(c) Flanking

(d) Aggression

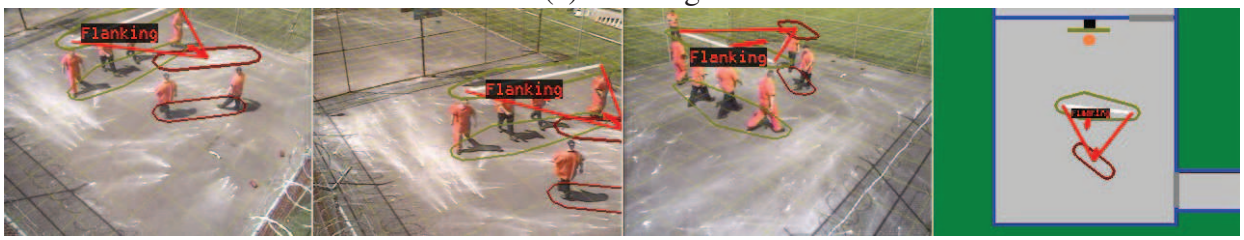
Figure 77: Two Groups Attacking Another. The images from top to bottom show the detection of (a) loitering, (b) distinct groups, (c) flanking, and (d) aggression. The shown activities have been enacted by law enforcement and corrections personnel.

enactment of the scenario.

Figure 79 shows a more complex scenario, lasting 3 minutes in which two gangs engage in an argument and after some discussion in separate corners of the recreation yard one gang decided to attack the other. The system again managed to detect the key components of this event and predicts the onset of the fight 1.5 seconds before it begins.

Figure 80 shows a comparison of the bottom-up MST grouping approach with the top-down modularity cut scheme. We found that the proposed modularity cut is superior in separating groups

(a) Flanking



(b) Aggression

Figure 78: Groups Attacking Another. The images from top to bottom show (a) flanking and (b) aggression detection. The shown activities have been enacted by law enforcement and corrections personnel.

during close interactions, an essential ingredient in analyzing small-scale changes in group structures. This encourages a further investigation into the use of our method.

The system presented here is able to perform the presented functions live and in real-time on a single quad-core workstation. In addition to video capture, processing and display it performed PTZ camera targeting and encoded all video to disk. The frame rate of the system typically varied between 22 Hz and 10 Hz, where most CPU cycles were consumed by the foreground-background segmentation and the feature tracker for the agitation detection.

C.7 Conclusions

This work aims at addressing the challenging problem of detecting suspicious and disorderly behaviors in complex environments where frequent social interactions occur. To tackle this challenge we utilize a sophisticated multi-camera multi-target tracking system that is able to track individuals even under crowded conditions. To establish an understanding of behaviors we perform a group level analysis of tracks of individuals. This allows the system to reason about events at a group level. In this particular work we presented a solution to detecting a variety of low level events of interest and in particular showed how the system is able to both predict as well as detect the onset of fights between groups of individuals. Future work will provide a more thorough quantitative

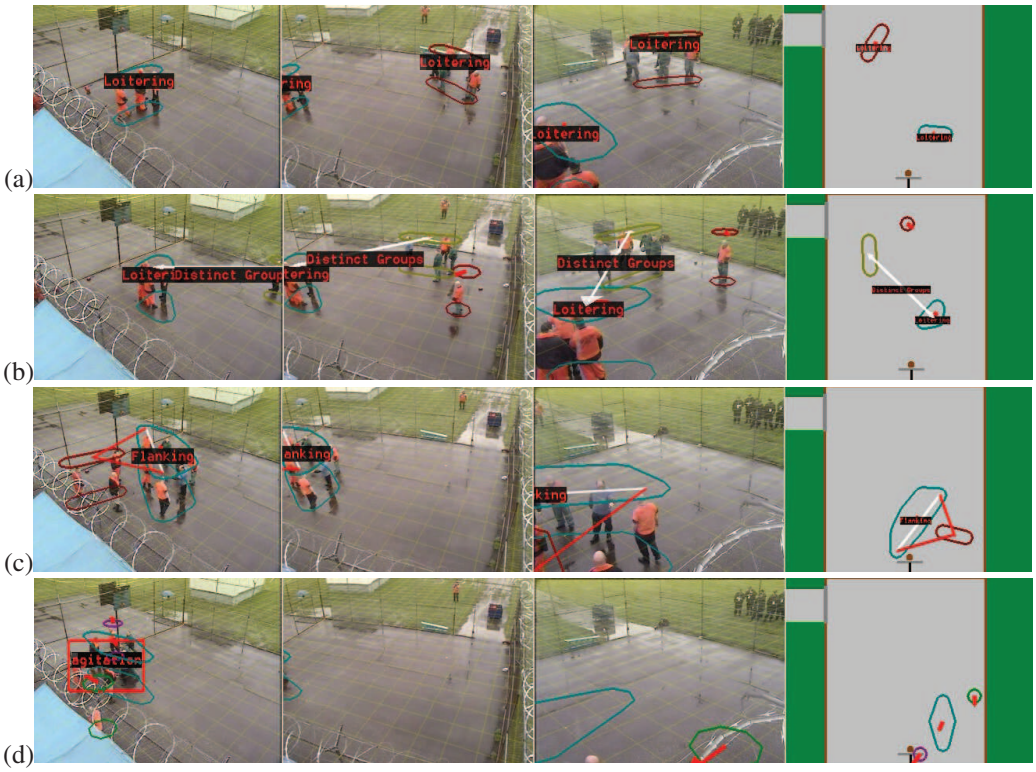


Figure 79: Large Group Attack. Detection of (a) loitering, (b) distinct groups, (c) flanking, and (d) aggression in a large complex interaction between two gangs. The shown activities have been enacted by law enforcement and corrections personnel.

analysis of the presented work and will investigate a probabilistic formulation of grouping and scenario-specific event detection.

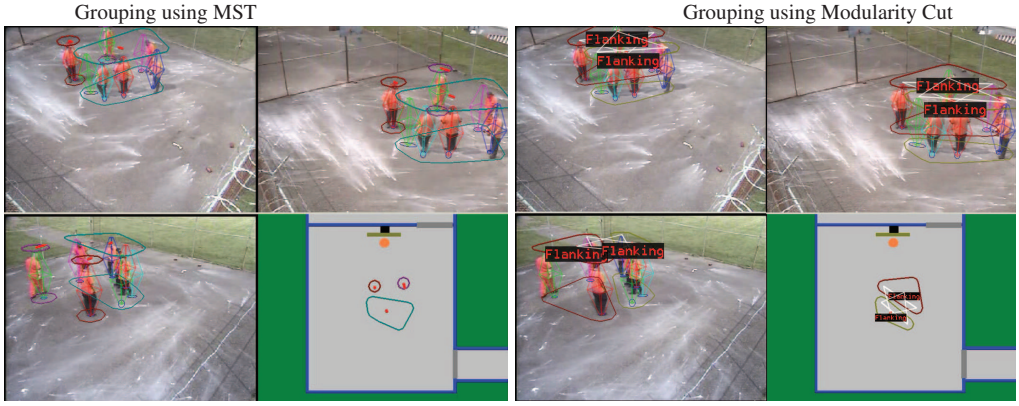


Figure 80: Flanking - Comparison of Results. This figure illustrates that in certain scenarios, the use of divisive clustering (modularity cut) is more advantageous than agglomerative clustering (MST), due to the ability of adaptive clustering. In this case, the MST clustering does not trigger a flanking event, while the modularity cut clustering correctly identifies the grouping structure and the event. The shown activities have been enacted by law enforcement and corrections personnel.

D Gaze Tracking

The following is a reprint of a paper that has been presented and published at the IEEE Workshop on Applications of Computer Vision (WACV), which took place in Kona, Hawaii on January, 2011. The precise title of the paper is: Karthik Sankaranarayanan, Ming-Ching Chang, Nils Krahnstoever, "Tracking Gaze Direction from Far-Field Surveillance Cameras", IEEE Workshop on Applications of Computer Vision (WACV), Kona, Hawaii, USA, Jan. 2011.

Abstract

We present a real-time approach to estimating the gaze direction of multiple individuals using a network of far-field surveillance cameras. This work is part of a larger surveillance system that utilizes a network of fixed cameras as well as PTZ cameras to perform site-wide tracking of individuals. Based on the tracking information, one or more PTZ cameras are cooperatively controlled to obtain close-up facial images of individuals. Within these close-up shots, face detection and head pose estimation are performed and the results are provided back to the tracking system to track the individual gazes. A new cost metric based on location and gaze orientation is proposed to robustly associate head observations with tracker states. The tracking system can thus leverage the newly obtained gaze information for two purposes: *(i)* improve the localization of individuals in crowded settings, and *(ii)* aid high-level surveillance tasks such as understanding gesturing, interactions between individuals, and finding the object-of-interest that people are looking at. In security application, our system can detect if a subject is looking at the security cameras or guard posts.

D.1 Introduction

Automatically understanding and recognizing behaviors from surveillance video in urban environments such as mass transit, schools and prison yards is challenging due to a large number of factors. Crowdedness and lack of resolution in a typical surveillance camera makes the accurate localization and tracking of individuals difficult and introduces uncertainty to subsequent reasoning stages. In such environment, one can at most hope to perform the localization of individuals

without further knowledge about body pose or orientation. Although body pose estimation has been studied in the context of surveillance [49], it is far from real-time performance. In this work we study part of the problem of estimating body pose, or more specifically head orientation of individuals in real-life videos, in order to: *(i)* reason over the orientation of individuals, in particular as part of pair-wise interactions between people, and *(ii)* to understand what people are looking at. The former feature is important for analyzing *group interactions* for which it is important to know *e.g.* if two people that are physically close, facing each other (mutual gaze), facing the same direction, or looking away from each other. The latter feature is of particular relevance to applications such as *retail security* [50] and *facility protection*, where security operators are interested in whether people are surveying (*i.e.*, looking at) camera locations, guard and clerk movements, or similar things of interest to a person who is about to commit a crime.

Head pose and gaze estimation from standard resolution surveillance views is challenging at best and impossible in many other cases. Hence, we utilize a hybrid approach where we perform multi-camera multi-target person tracking within a network of fixed cameras, and pass the tracking information to drive one or more PTZ cameras to zoom-in on individuals (detailed in §D.3). Face tracking and head pose estimation is then performed within the close-up views of these PTZ captures, fusing information hand-in-hand with the person tracker. The face location and head pose information is mapped back into an unified coordinate system, where it can be used to improve the tracking performance (under crowded conditions) and perform higher level reasoning over the pose (*e.g.*, to analyze social interactions [40] or behaviors [51]). The proposed system operates in real-time under challenging imaging conditions. Due to the relatively larger computational burden in face detection, the person tracking and face tracking must operate *asynchronously*. We will elaborate in §D.4 how we integrate the information dynamically and consistently in a flexible framework.

This work is the first (to the best of our knowledge) that investigates the augmentation of multi-camera tracking with multi-PTZ facial gaze tracking in the surveillance domain. Our main contribution is a unified approach to robustly fusing together person tracking information with

asynchronous PTZ facial tracking information. Our system is flexible to operate on either a single or multiple cameras. While the use of multiple cameras is not a hard requirement, it does improve the overall tracking performance, in particular in situations where multiple PTZ cameras view a group of people from a set of different directions.

The paper is organized as follows. We will describe related work in §D.2, the overall system in §D.3, and our approach to gaze analysis in §D.4. We will present real-time experimental results in a variety of settings in §D.5. We will discuss the results in §D.6 and conclude the paper in §D.7.

D.2 Related Work

Head pose estimation from one or more views has been extensively studied over the past 15 years with applications ranging from robotics, human computer interaction [52], driver assistance, and virtual reality. The recent review article of Murphy-Chutorian and Trivedi [53] provides an excellent summary and comparison between various approaches including using appearance, non-linear regression, non-rigid model fitting, tracking and hybrid methods. Among them we highlight automatic methods that detect and track head pose from single or multi-view videos in an unconstrained environment. Works in this category [54, 55] involve head detection followed by pose estimation and tracking, *e.g.* using Kalman filter [56] or particle filtering [57, 58].

Hu *et al.* [59] fit a single Active Appearance Model (AAM) simultaneously to multiple synchronous face images to estimate head pose, with a requirement that the head image quality must be high enough. Voit *et al.* [60] use a neural network classifier to estimate head pose from each view and use Bayesian dynamics to merge estimations, with a strong assumption that the individuals are sitting in fixed seats such that no tracking or camera zooming is required. Lanz and Brunelli [58] track body parts using a Bayesian framework over shape and appearance and estimate head orientation across multiple views using particle filtering. Canton-Ferrer et al. [61] assume head location is known and estimate its orientation by back-projecting the skin appearance patches onto the estimated 3D head model and employ a particle filter in tracking across multiple views. In a recent work, Bäuml et al. [57] assume head location is known and track face pose across a dis-

tributed camera network for recognition and re-identification. The face tracker runs separately and independently from the person tracker, and there is no attempt to exploit the advantage of multiple overlapping facial views.

To the best of our knowledge, all existing works make strong assumptions that *(i)* the head locations are (roughly) known and *(ii)* head image quality is (reasonably) good, so as to simplify the problem of simultaneous tracking and pose estimation. A major difference that sets the proposed work apart is that our system operates in a more unconstrained, challenging environment in live, where both person locations and head poses are unknown, in addition to that the close-up PTZ views are dynamically changing as well. The person tracking and PTZ face tracking thus must be performed asynchronously. We try to bring together various observations and fuse them into a consistent, central tracking scheme (see §D.4).

D.3 System Description

In this section we will provide a brief outline of our tracking system as well as the type of environment we are addressing in this paper.

D.3.1 Video Tracking System

The tracking system that we utilize [38, 36, 40] comprises of multiple calibrated static cameras tracking cooperatively in a synchronized fashion. For each view, the position and image dimension of each person at all possible 3D locations in the scene are estimated using calibration. Foreground pixels from online tracking are used to vote for these precomputed image locations to form a set of (foreground) detections [38]. This effectively leverages the calibration information to significantly reduce false positives arising from occlusions and crowdedness.

The set of detections for each view are then projected onto the ground plane in 3D in order to further disambiguate any confusion due to occlusions and crowdedness. These projections are consumed by a centralized tracking system that either *(1)* associates detections with existing tracks based on spatial proximity or *(2)* initiates new tracks. The states of tracks are estimated by a

standard Kalman filter, performed in the world reference ground plane. The system is designed to maintain tracks across camera boundaries in order to perform site-wide tracking.

D.3.2 Pan Tilt Zoom Control

To enable face detection and gaze estimation of uncooperative individuals from a distance, the tracking system controls multiple pan tilt zoom (PTZ) cameras automatically [62]. The control algorithm pursues the goal of optimally scheduling the PTZ cameras in real-time under a variety of performance objectives. The control system provides each PTZ camera with a continuously evolving *schedule* that describes what targets to visit in what order. Schedules are planned several target capture steps into the future based on the current and predicted motion of observed individuals. A given schedule is assigned a probability of achieving the goal of capturing high quality facial shots of all tracked individuals. The quality of facial shots is governed by the distance of individuals from the camera, the angle at which a face is captured, and the accuracy with which a person is being located by the tracking system. A control strategy is chosen by selecting the schedule with the highest probability from the set of all possible schedules. Details of our PTZ control approach are provided in [62].

Contrary to the fixed cameras, the poses of PTZ cameras change over time. As part of the control, the system's task is to estimate the time varying projection matrices for the PTZ cameras, given as $\mathbf{P}_i^{\mathrm{p}}[t]$, where $i \in \{1, \ldots, N^{\mathrm{P}}\}$ the index over all PTZ cameras and t the time.

In the majority of the experiments shown below we utilize a testbed that is equipped with four fixed cameras for tracking and four PTZ cameras for face capture. It should be noted that the ratio between the number of tracked individuals and available PTZ cameras makes an impact on system performance. A larger number of individuals typically requires a larger number of PTZ cameras or the utilization of high resolution mega-pixel cameras.

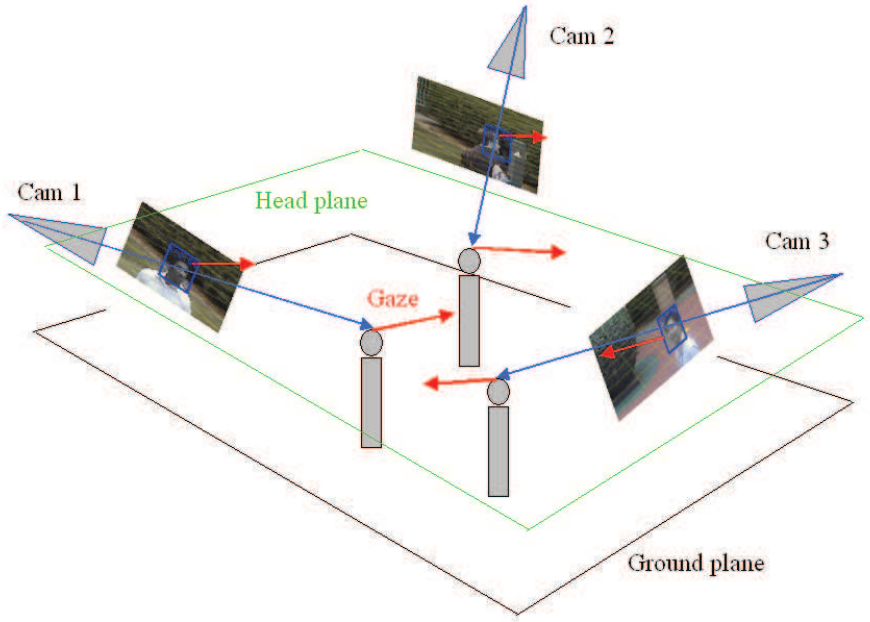


Figure 81: Projecting the detected faces to the head plane and calculating the gaze vectors.

D.4 Gaze Analysis

In this section, we describe in detail our method to obtain gaze tracks for each of the individuals using a Kalman filter based gaze orientation tracking system. We assume that a person's head pose generally aligns with one's gaze direction, even though the head pose is only a coarse estimate of visual gaze (*i.e.* eye ball) direction [53].

As the PTZ cameras locate the individuals, they zoom into the estimated head locations, such that face detection can be performed in each PTZ views. The detected face locations are projected back to the 3D head-plane to obtain an estimate of the person's head position in the 3D world (§D.4.1). Meanwhile, the head pose is estimated from the face image and converted to a 3D gaze vector using the PTZ camera's rotation matrix (§D.4.2). This is performed for each PTZ camera that is currently obtaining individual head/gaze location and orientation, see Fig. 81. We develop a Kalman filter based gaze tracker that operates on angular coordinates of the gaze vectors. The tracked gaze orientation augments the person tracker (which is in fact another Kalman filter tracker) that operates on the ground plane. We utilize the Hungarian algorithm [63] to associate

the location and orientation of the faces to the individuals by minimizing a cost function (§D.4.3). Note that our state and observation spaces are both in angular coordinates so there are no non-linearities involved. We track transformed observations rather than raw observations that would be non-linearly tied to the state space. Once the observations corresponding to different trackers are obtained, a Kalman filtering update is performed.

D.4.1 Face Detection and Projection

We use an off-the-shelf face detector [64, 65] to detect faces in the PTZ views. The algorithm is chosen because it works well with a wide variation in head poses, from frontal to profile views. It is able to detect faces in fairly low-resolution video. In this work, we deal with 640×480 pixel images and the system controls the PTZ to capture facial shots with a rough resolution of 20-30 pixels eye-to-eye. The face detector has a low false-positive rate in our system. As we will demonstrate later, remaining false-positive detections can be handled robustly by the gaze tracker.

Face detections in each image view are used to estimate each individual's head location in the 3D world. This is done by *(i)* projecting a ray from the optical center of the PTZ camera $\mathbf{P}_i^{\mathrm{p}}[t]$ through the center of the face location in the image plane, and then *(ii)* finding the intersection of this ray with the head-plane, which is assumed parallel to the ground plane at a height of 1.8 meters; see Fig. 81. Also, the width and the height of the face are used to estimate a covariance confidence level for the face location. The covariance is projected to the groundplane using an unscented transform (UT) from the image to head plane, followed by downprojection to the groundplane. The above operation is performed for all face detections in all PTZ views to obtain multiple head locations for all individuals, with estimates of (mean, covariance) pair simultaneously. All observation information is organized in a unified 3D world coordinate system, where a central tracker can operate in an integrated manner.

Along with the face detection, face orientation (head pose) can be estimated from either using *(i)* active appearance models (AAM) matching [66, 67], or *(ii)* face feature detection followed by pose estimation (as done in [65]). In theory, one could model the full egocentric parameters of the

head: the *yaw* (left/right), *pitch* (chin up/down) and *roll* (around "nose" axis). However the roll direction is not significant for gaze estimation, and the pitch is often unreliable. We thus mainly focus on tracking the yaw orientation. As we will show in §D.4.2, we can still estimate the *global* head pitch, because its 3D pose is viewed from different PTZ cameras mounted at different heights.

D.4.2 Head Pose to 3D Gaze

In order to track gaze from multiple cameras, we need to transform it from local camera coordinates to the central 3D world space (to be tracked by the central tracker). To do this, the gaze vector (face normal) is first obtained in Cartesian coordinates in the camera space from the head pose angles, and transformed to the world space using the camera rotation matrix. Finally the transformed gaze vector is converted back in terms of egocentric angles (orientations) in the world space.

To first obtain the face normal, that is the gaze vector local to the camera coordinate system $\mathbf{g}_{im}$=(x_{im}, y_{im}, z_{im}) from the head pose yaw angle (ϕ_{im}), we use the following equations:

$$x_{im} = \cos(\phi_{im}), \quad y_{im} = \sin(\phi_{im}), \quad z_{im} = 0. \tag{74}$$

The rotation matrix of the PTZ camera is then used to convert the gaze vector from the local image space to the 3D world space. Using the technique of transforming normals, the gaze vector is multiplied by the transpose of inverse of the rotation matrix of a PTZ camera $\mathbf{P}^{\mathrm{p}} = [\mathbf{R}|\mathbf{t}]$ (we will ignore the PTZ index i in the following):

$$\mathbf{g}_w = \mathbf{g}_{im} * (\mathbf{R}^{-1})^{\mathbf{T}}. \tag{75}$$

The transformed gaze vector $\mathbf{g}_w = (x_g, y_g, z_g)$ is converted to back to the egocentric angles repre-

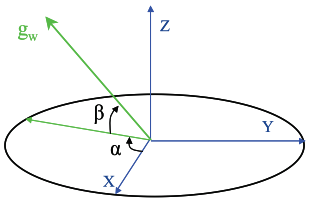


Figure 82: Relation of the gaze egocentric angles (yaw α and pitch β) with the gaze vector g_w.

sentation of yaw (α) and pitch (β) in 3D space (see Fig. 82) using the following equations:

$$\alpha = \arctan(\frac{x_g}{y_g}), \tag{76}$$

$$\beta = \arctan(\frac{z_g}{\sqrt{x_g^2 + y_g^2}}). \tag{77}$$

At the end of this step, the head location and gaze orientation is obtained for one or more targets from multiple PTZ cameras projected to a common centralized 3D space.

D.4.3 Kalman Filtering for Gaze

In order to track the gaze orientations of the individuals, we extend the person tracker state by adding gaze information to it. The gaze state of a target (Θ) is modeled in terms of the two orientation angles (α, β), as well as their first derivatives — the angular velocities ($\dot{\alpha}, \dot{\beta}$). Therefore, $\Theta = \begin{bmatrix} \alpha & \beta & \dot{\alpha} & \dot{\beta} \end{bmatrix}^{\mathbf{T}}$.

The state transition model that relates the gaze state at time $k-1$ to the state at time k is given as

$$\Theta_k = \mathbf{F} * \Theta_{k-1} + \mathbf{w}_{k-1}, \tag{78}$$

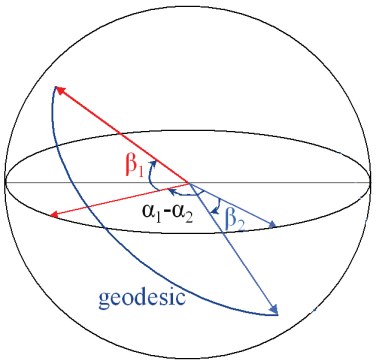


Figure 83: Geodesic distance between orientation vectors measured on the great circle.

where the state transition matrix ($\mathbf{F}$) using a constant velocity model is given as

$$\mathbf{F} = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{79}$$

and $\mathbf{w}_{k-1}$ is the gaussian distributed process noise. The measurement model which extracts the orientation information from the gaze state (Θ) is given as

$$\begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} * \Theta_k + \mathbf{v}_k, \tag{80}$$

where $\mathbf{v}_k$ is the gaussian distributed measurement noise.

D.4.4 Data Association for Faces

Since we are dealing with the tracking of multiple individuals, a necessary step is to associate the different face detections obtained from multiple cameras with their corresponding trackers so that the gaze states of the trackers can be updated appropriately. Note that even though the system knows the identity of an individual when the PTZ camera is allocated to look at, it is still possible that multiple face detections are obtained from a video frame, especially when individuals are

close to each other. Therefore, the system may not know exactly which face in the image belongs to which individual. Consequently, this data association step becomes essential to resolve the ambiguities in assigning gaze detections to tracks reliably.

The data association module uses two cues to assign the detected faces to trackers appropriately: *(1)* Head location (from §D.4.1) and *(2)* 3D gaze orientation (from §D.4.2).

Let N_d be the number of face detections and N_t be the number of trackers at a given timestep k. In order to perform the data association, we need a distance metric to measure the distance between head observations $\mathbf{h}_i$ (where $1 \leq i \leq N_d$) and tracker states $\mathbf{t}_j$ (where $1 \leq j \leq N_t$). The following cost metric η is proposed to measure the distance between head observation $\mathbf{h}_i$ and a tracker state $\mathbf{t}_j$.

$$\eta(\mathbf{h}_i, \mathbf{t}_j) = \exp(-\frac{d(\mathbf{h}_i^{\mathbf{x}}, \mathbf{t}_j^{\mathbf{x}})}{\sigma_{\mathbf{x}}} - \frac{\lambda(\mathbf{h}_i^{\Theta}, \mathbf{t}_j^{\Theta})}{\sigma_{\Theta}}), \tag{81}$$

where $d(\mathbf{h}_i^{\mathbf{x}}, \mathbf{t}_j^{\mathbf{x}})$ is the Euclidean distance between the head observation's location on the head plane and tracker's location on the ground plane (ignoring the height difference). $\lambda(\mathbf{h}_i^{\Theta}, \mathbf{t}_j^{\Theta})$ is the geodesic distance (see Fig. 83) between the gaze orientation of the head observation and the tracker's current gaze orientation, which is calculated using the spherical law of cosines as

$$\begin{aligned}\mathbf{A}_{ij} = \lambda(\mathbf{h}_i^{\Theta}, \mathbf{t}_j^{\Theta}) = \arccos(&\sin\beta_{\mathbf{h}_i} \sin\beta_{\mathbf{t}_j} \\ &+ \cos\beta_{\mathbf{h}_i} \cos\beta_{\mathbf{t}_j} \cos(\alpha_{\mathbf{h}_i} - \alpha_{\mathbf{t}_j})).\end{aligned} \tag{82}$$

Using the above cost function, the distance between every pair of the i-th head detection and the j-th gaze track is calculated to build a cost matrix $\mathbf{A}_{ij}$ of size $N_d \times N_t$. The task of assigning the heads to the correct trackers is now a combinatorial optimization problem. For this, the Hungarian algorithm [63] is employed to find an optimal assignment of observations to trackers (by minimizing the cost) in polynomial time.

Once the faces have been assigned to their respective trackers, the standard update step of the Kalman filter is performed to update the gaze state of each individual being tracked. In order

Figure 84: **Sequence A:** Face detections and head pose estimation from three PTZ camera views (top) projected to the three corresponding static camera views (bottom). Observe how qualitatively the gaze vectors are tracked from the visualization of a few trailing frames around the subject. Yellow mesh grids visualize the ground plane in projective views.

to visualize the gaze of the target at every point during tracking, the gaze angles (α, β) from the state vector of the Kalman filter are converted into their corresponding Cartesian representation as follows:

$$\hat{x}_{gaze} = \frac{\cos\alpha}{|\cos\alpha|}, \tag{83}$$

$$\hat{y}_{gaze} = \hat{y}_{gaze}\tan\alpha, \tag{84}$$

$$\hat{z}_{gaze} = \sqrt{\hat{x}^2_{gaze} + \hat{y}^2_{gaze}}\tan\beta. \tag{85}$$

D.5 Experiments and Results

We first demonstrate results from the first part of the system, which is the face detection and gaze vector calculation from head poses. After that we perform experiments and demonstrate results with tracking the gaze of one or more individuals.

Figure 85: **Sequence B:** Face detections and gaze tracking for multiple individuals in the scene. Observe the asynchronous nature of the mixture of the person tracking (diamond outlines) and face detection updates (grayscale boxes), suggesting the need for a robust data association.

D.5.1 Gaze Observations from Head Pose

Our system consists of four fixed cameras and four PTZ cameras overlooking portions of a courtyard. As the individuals walk around, the fixed cameras are used to perform tracking and the PTZ cameras zoom into the calculated head location and performed face detections. These detections are then used to estimate the gaze vectors. Results from test sequence A are shown in Fig. 84. Face detections from the PTZ cameras provide different views and consequently different head poses. These poses are transformed to obtain 3D gaze vectors, which are visualized back in the static views. In the bottom row, different colors of the vectors correspond to gaze vector coming in from different cameras. Also shown are gaze vectors from a few trailing frames with reducing intensity.

Fig. 85 shows a few frames of another test sequence, where multiple individuals are tracked and their face detection are associated to form gaze tracks. Even though PTZ cameras are allocated to particular targets, multiple face detections are obtained from each view. Consequently, the association of individual face detections to from gaze tracks becomes necessary (§D.4.4).

Figure 86: **Sequence A:** Few frames of simultaneous person tracking and gaze tracking at different time.

D.5.2 Gaze Tracking

Single-person gaze tracking: The gaze observations from previous section are used to update the state of the Kalman filters corresponding to each individual's gaze tracker. The gaze orientation vectors are then projected onto each camera views for visualization. Figs. 86 and 88 show such visualization from a few frames of gaze tracking in sequences A and C, respectively. Fig. 87 plots the Kalman filter states α and β for sequence A against time, as well as a scatter plot of the two, demonstrating a smooth tracking of the target's gaze orientation.

Multi-person gaze tracking: We also performed experiments on sequences with multiple individuals in the scene. Fig. 89 (top) shows results of gaze tracking from two interacting individuals. Fig. 89 (bottom) shows the tracking of three individuals.

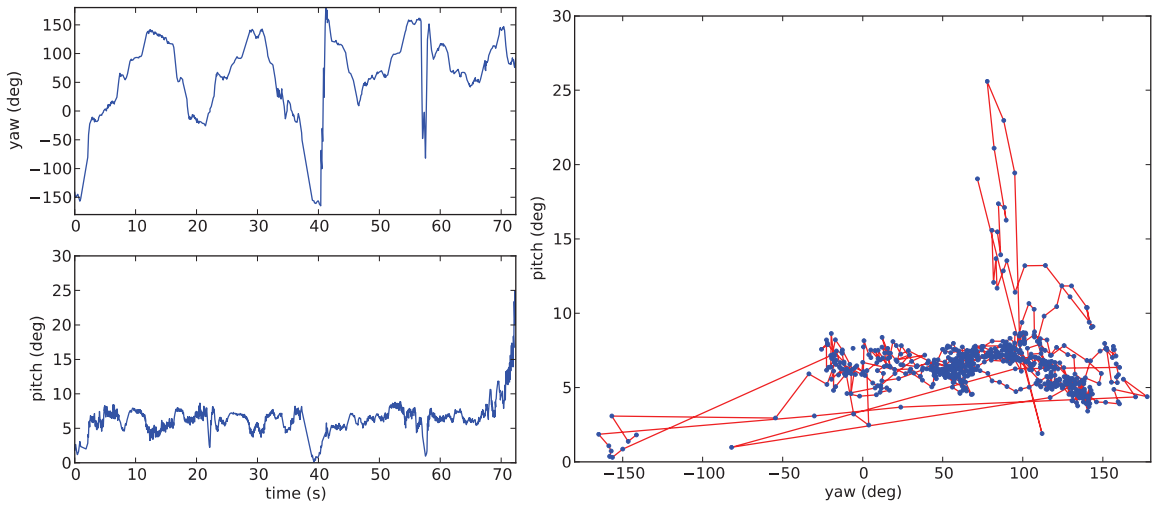


Figure 87: **Sequence A:** (Left) Gaze trajectory in yaw (α) and pitch (β) plotted against time. (Right) Scatter plot of yaw vs. pitch.

D.6 Discussion

D.6.1 Improving primitive surveillance tasks

Gaze tracking has a lot of potential to improve primitive video surveillance tasks like person detection and tracking. *(i)* For example, the high-res face location can be used to improve person tracking accuracy. As shown in Fig. 90, when the tracker (red diamond) starts to deviate away from the actual target location, at this point the face detection location on the head plane can be treated as a new target observation and correct the tracker location estimate. *(ii)* Similarly, the gaze can also be used to predict the future locations of the person, under the assumption that in most cases a person walks in straight direction one is looking. *(iii)* The gaze states of the trackers can also be helpful in circumventing common issues like trackers getting switched between individuals that are standing close to each other. This can be done by looking at the gaze states of both trackers and using that to resolve the ambiguities. This is especially important in any system that looks to perform group behavior analysis.

Figure 88: **Sequence C:** Few frames of simultaneous person tracking and gaze tracking at different time.

D.6.2 Potential Applications

Surveillance: The proposed multi-view multi-target gaze tracking system has application in behavior and social group analysis. Gaze provides valuable information to aid detecting events such as grouping formation, aggression [51]. More importantly, it may provide cues towards prediction and prevention of harmful events.

Retail application: The proposed system has applications in retail that the gazes of customers can be studied to infer product preferences. This system could also be used to study the reaction of customers to advertisements to study attention characteristics [50].

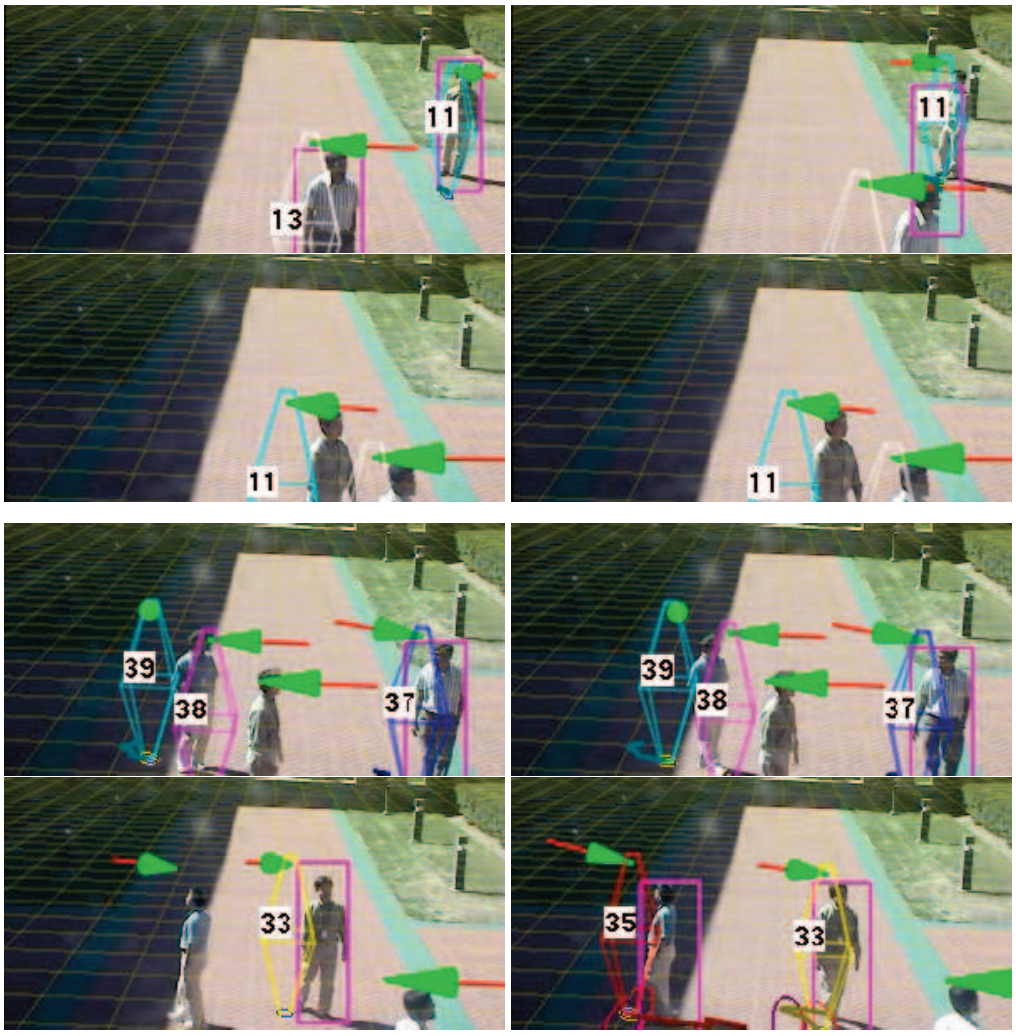


Figure 89: **Sequences D, E:** Gaze tracking with multiple individuals in the scene. Top two rows: two individuals. Bottom two rows: three individuals.

D.6.3 Challenges

A major challenge with tracking gazes of multiple targets in any environment is the limitation of number of gaze observations that are obtained. This is a hardware issue and is a function of the number of cameras installed. Nowadays, with the proliferation of cameras and their reducing costs, this issue can be expected to of lesser importance if not disappear in the future. Another challenge in real-time performance is the difference in running speed of the face detection module as compared to the ground plane tracking, which could possibly result in a few seconds lag between the face detections and the tracking. We consider this to be a factor of hardware and expect this

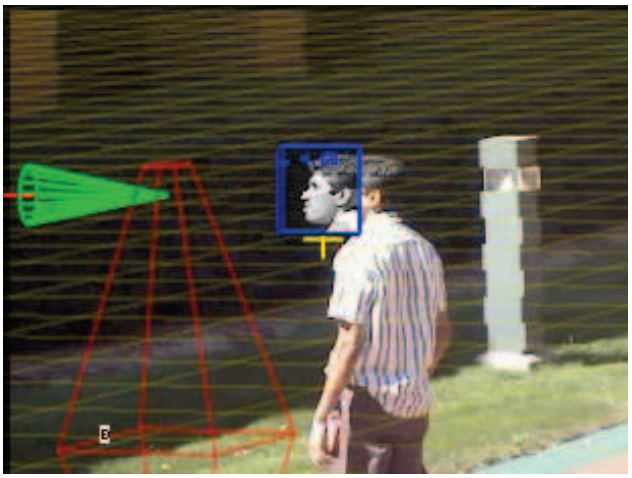


Figure 90: Example of where an individual's ground plane location estimation can be improved by using the observation from face detection.

concern to be alleviated with faster systems in the future. The head pose estimation obtained in our system is relatively coarse grained and can be improved with better pose estimation algorithms (*e.g.*, [66] with a tradeoff in speed), with the requirements decided by the application.

D.7 Conclusions

We have presented a multi-PTZ, multi-target gaze tracking system that operates in real-time in unconstrained environments. A gaze vector is estimated for each individual based on face detection and a head-plane back-projection. A centralize Kalman filter tracking system is implemented to model the gaze tracks. A new cost metric based on location and gaze orientation is also proposed to robustly associate head observations with tracker states. Experimental results with multiple sequences demonstrate potential in surveillance, security, retail, and social group analysis applications.

Future work includes a quantitative validation of gaze tracking accuracy and reliability using standard datasets summarized in [53].

E Probabilistic Group Analysis

The following is a reprint of a manuscript that is going to be presented and published at the International Conference on Computer Vision (ICCV) 2011. The precise title of the paper is: Ming-Ching Chang, Nils Krahnstoever, and Weina Ge, "Probabilistic Group-Level Motion Analysis and Scenario Recognition", International Conference on Computer Vision (ICCV), Barcelona, Spain, Nov. 2011.

Abstract

This paper addresses the challenge of recognizing behavior of groups of individuals in unconstraint surveillance environments. As opposed to approaches that rely on agglomerative or decisive hierarchical clustering techniques, we propose to recognize group interactions without making hard decisions about the underlying group structure. Instead we use a probabilistic grouping strategy evaluated from the pairwise spatial-temporal tracking information. A path-based grouping scheme determines a soft segmentation of groups and produces a weighted connection graph where its edges express the probability of individuals belonging to a group. Without further segmenting this graph, we show how a large number of low- and high-level behavior recognition tasks can be performed. Our work builds on a mature multi-camera multi-target person tracking system that operates in real-time. We derive probabilistic models to analyze individual track motion as well as group interactions. We show that the soft grouping can combine with motion analysis elegantly to robustly detect and predict group-level activities. Experimental results demonstrate the efficacy of our approach.

E.1 Introduction

Environments such as schools, transportation hubs, sport venues, and public gatherings are typically characterized by a large number of people that exhibit frequent and complex social interactions [68, 69]. In order to identify activities and behaviors in such environment, it is necessary to understand the interactions taking place at a group level [70, 30, 71]. Understanding group-level interaction is particularly important in surveillance and security, where the gang related activities

are the root cause of most criminal behaviors and disorderly conduct. A major goal of this work is to automatically detect and predict events of interest by understanding behaviors and activities at the group level.

There are at least three major issues in performing group-level behavior recognition. First, one needs to define the group structure from a varying number of individuals. Existing methods that track this group structure for behavior recognition [19, 72, 31, 71, 30, 73, 32] mostly rely on a hard decision over an explicit grouping. In our experience this leads to brittle reasoning systems that are sensitive to noise and tracking inaccuracies. Second, the spatial-temporal relationships among individuals can change rapidly, especially in busy environments, which poses the challenge of maintaining and tracking evolving group structures. Approaches based on hard grouping lack the capability to deal with ambiguity in grouping, which are often observed during a transition period when a person gradually joining or leaving a group. Third, given the temporally evolving group structures, efficient inference strategy needs to be investigated in order to perform event recognition.

To the best of our knowledge, our work is the first of the kind to perform and maintain a **soft** grouping structure throughout the entire event recognition stage. The main contribution of this paper is a probabilistic approach to both determine the group structure and perform robust reasoning on top of it. A key feature of our soft grouping strategy is that group-level activities must be represented on a per-individual basis. Our framework can probabilistically predict motion patterns and group-level activities of interest, such as "Are individuals i, j, and k forming a new group?" and "Are two groups going to meet in the future?". Our system operates based on two components:

- a *probabilistic group analysis* to reason about the soft group structure between individuals based on a connectivity graph defined using a track-to-track and a path-based connectivity measure.
- a *probabilistic motion analysis* to reason over the spatial-temporal pattern both at an individual and a group level to perform scenario recognition.

Our approach is capable of handling arbitrary number of individuals. The group representation is general and can be combined naturally with subsequent reasoning — analytic rules can be motivated directly from its probabilistic formulation in combining with other event inference modules. Our recognition framework is thus flexible in adapting new scenarios. Moreover, our model construction is intuitive (user-friendly for non-technical operators) and invariant to site-specific observations. This is because we construct group activity models using scenario-specific predicates. We did not take a learning approach [74] for two reasons: (1) When data availability is limited, the learning of event models is difficult and requires considerable expertise. (2) The integration of domain knowledge in the model design phase is important. Our approach enables the end user to quickly design a new scenario recognition module based on combining existing modules.

This paper is organized as follows. §E.2 covers related works. §E.3 describes our probabilistic group analysis. §E.4 describes how we combine the soft group analysis and track motion analysis (which will be elaborated in §E.5) to recognize group-level activities. §E.6 describes the application domain with performance validation. §E.7 concludes the paper.

E.2 Related Works

Crowd behavior recognition has been an active research area [68, 69]. We briefly review works that perform group tracking in particular for event recognition. Saxena *et al.* [32] detect abnormal crowd events by tracking feature points using a KLT tracker and building a scenario recognition engine based on thresholding motion measurements. Hoogs *et al.* [71] detect crowd formation and dispersion via relational clustering solved using a spectral graph clique analysis. Activity models are manually specified a priori and limited to a single constant temporal state. Ge *et al.* [31] identify small groups of crowd to study pedestrian behavior. Groups are formed using bottom-up hierarchical clustering with hard decisions.

Ni *et al.* [70] recognize human group activities using three types of localized causalities. Their approach is based on learning concerning training samples and data labeling and is not automatic in requiring manual initialization of human tracks. Ryoo and Aggarwal [73] simultaneously estimate

group structure and detect group activities using a stochastic grammar. Grouping is determined by reasoning over specific spatial-temporal relations. Lau *et al.* [30] estimate group structure without identifying individual blobs in tracking. Both the detection-track association and the grouping are hypothesized and posed as a recursive multi-hypothesis model selection problem. Although probabilistic clustering is performed, hard decisions are made in determining the group size.

For pedestrian tracking and motion prediction, most existing methods focus on modeling simple activities of a single person or the interactivity between two. In contrary, activities or interaction among groups, which occur more often in real scenarios, are much less devoted. Existing methods include using social force model [75, 76], floor field [77], correlated topic model [78], and use multi-camera tracking [79, 80]. Abnormal activities are detected using motion pattern [72], bag-of-social force features [76], mixture of dynamic textures [81], *etc.* The flow field tracking of Shah *et al.* [77, 76] handles very crowded scenes; however it does not operate at a group level.

E.3 Probabilistic Group Analysis

Defining a precise *grouping* of a crowd is challenging due to the complex social interactions and relations that are hard to measure. To handle the uncertainty in video tracking, we avoid explicit group segmentation and instead maintain a probabilistic measure.

E.3.1 Pairwise Grouping Measure

We first seek an instantaneous pairwise group affinity measure (that represents the probability of a pair of people belonging to a group), by checking if two individuals are physically close. Inspired by standard social norms from Hall's *proxemics* theory [2] for modeling inter-person spacial relations and the social force model [3] for modeling pedestrian dynamics, we define a pairwise grouping measure based on three main terms: the *distance* between two individuals, the *motion* (body pose and velocity) and the *track history*, as illustrated in Fig.91. How this direct connection probability can be extended to express group membership of non-direct neighbors will be described in §E.3.2.

The above pairwise grouping measure is defined straight from track observations, thus it favors people that are spatially close. Consider that affinity between people is not always isotropic, our individual-centric affinity is not so, either. Denote d_{ij} the Euclidean distance between two people i and j located at $\mathbf{x}_i$ and $\mathbf{x}_j$ on the ground plane (Throughout the paper, symbols i, j, k refer to a track of a person). We assume a person always faces one's motion direction and denote with ϕ_{ij} the angle between i's velocity vector and the relative position vector $\mathbf{p}_{ij} = \mathbf{x}_j - \mathbf{x}_i$ *w.r.t.* person j. In addition, the affinity varies with the velocity magnitudes of i and j. Overall the instantaneous affinity measure for some time t is hence a function

$$p_c^{\text{inst}}(i,j) = f(d_{ij}, \phi_{ij}, ||v_i||, ||v_j||), \tag{86}$$

where subscript c stands for connectivity, and the dependency of time t is made implicit for clarity. Fig.91(d) visualizes our concrete measure hidden behind the abstract definition (Eq.86). Notice the probability is higher on the side of a person than in the front or back, which is a direct implementation of the aforementioned social norm that states people in a group are more likely to walk side-by-side.

To incorporate track history for robust estimation, we take account p_c^{inst} at time t over a window of T seconds (*e.g.* $T = 3$):

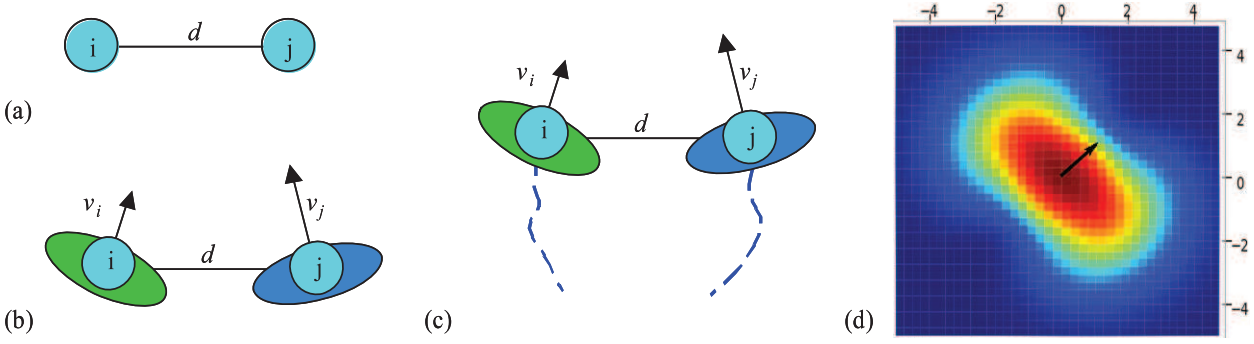


Figure 91: Pairwise group affinity measure: (a) Inter-person distance. (b) Distance and motion (velocity magnitude, direction, frontness/sidedness). (c) Distance, motion, and track history. (d) The instantaneous affinity measure of an individual at $(0, 0)$ with velocity vector $(1, 1)$ in an arrow. Color map depicts this probability kernel between 0 (blue) and 1 (red).

$$p_c^p(i,j;t) = \omega_1 \;\; p_c^{inst}(t) + \omega_2 \;\; \frac{\sum_{t_i \in T} p_c^{inst}(t_i)}{|t_i \in T|}, \tag{87}$$

where ω_1, ω_2 adjust the weights between the two terms of current status and the entire window history ($\omega_1 + \omega_2 = 1$). This improves overall robustness and avoid treating a sudden "passing by" event as an abrupt group change.

E.3.2 Path-based Group Connectivity

The pairwise affinity measure $p_c^p(i,j;t)$ is defined for two individuals i, j, independent of all other people in the crowd. Observe that two arbitrary individuals in a group do not necessarily have to be directly connected. Rather, it is sufficient that a connecting chain of bonds exists. Here we introduce a path-based group connectivity that estimates the pairwise grouping probability under the influence of others. We say that i and j are *connected*, if there exist pairwise connected intermediate individuals $i_0, ..., i_N$:

$$p_c^\pi(\{i \text{ and } j \text{ are connected via } i_0, ..., i_N\}) = \;\; p_c^p(i, i_0) \left[\prod_{k=0}^{N-1} p_c^p(i_k, i_{k+1}) \right] p_c^p(i_N, j). \tag{88}$$

We then set the connection probability between i and j to be the optimal path amongst all possible paths, which yields the highest probability:

$$p_c^\pi(\{i \text{ and } j \text{ are in same group}\}) = \max_{\text{all paths } P_k} p_c^\pi(\{i \text{ and } j \text{ are connected via path } P_k\}). \tag{89}$$

To find the optimal path, we first define the edge weight of the initial connection graph to be $\mathbf{G_0}(i,j) = -\log(p_c^p(i,j))$, whose values ranges from 0 to ∞. We then use Floyd's algorithm [15, Ch.32] to compute the all-pair shortest path in $O(n^3)$, where n is the number of tracked individuals. The resulting graph $\mathbf{G_0^\pi}$ contains non-negative path weights. We then obtain the final probabilistic connection graph by $\mathbf{G} = p_c^\pi(i,j) = \exp(-G_0^\pi(i,j))$, where $p_c^\pi(i,j)$ is the path-based grouping probability.

The intuition behind this is that the grouping of (i,j) should directly depend on the path cre-

ated by other individual k in between them. Our path-based metric could be viewed as a simplified solution of a more sophisticated flux-based model, where the connectivity between all pairs of individuals is formulated as a flow, and consider the accumulated flux as the grouping connectivity using the standard *maximum-flow, minimum-cut* algorithm [13, Ch.27]. However the computational cost for the flux-based metric is high (exponential). Our algorithm is also inspired by the spectral clustering [16] and path-based clustering [17] in the domain of pattern classification [18, Ch.10.9].

In case an explicit grouping is desired, we can adopt a proper graph cutting method on $\mathbf{G}$ such as using the hierarchical agglomerative clustering (*e.g. minimum spanning tree* (MST) [19]) or *modularity-cut* [20]. Fig.93 visualize $\mathbf{G}$ (in transparent edges) as well as some explicit grouping (in color polygons) in our test scenarios. Group segmentation is more robust if the hard decision is made only at the last stage of grouping. Our path-based grouping is less bias than the MST-grouping, since all pairs of paths are considered, whereas in [19] weaker connectivity are ignored in the clustering process.

We will show in the next section that we can perform many reasoning tasks using $\mathbf{G}$ without explicit grouping: counting the number of individuals in a group, determining if a group is forming or dispersing, modeling the movement of a group, and at a high level if separate groups are about to engage in aggressive activities such as a fight — all using similar probabilistic reasoning steps.

E.4 Probabilistic Group Structure Analysis and Scenario Recognition

We describe the key concept toward the flexibility and robustness of our approach, that is to *represent and reason group-level activity on an individual basis using the soft grouping graph* $\mathbf{G}$. This is a novel perspective because no decisive grouping is performed during the reasoning process. Since no explicit grouping is made, we must define the probability of a group-level scenario on an individual basis. For example, "the probability of the group that person i belongs to is chasing the group of j is 0.3". Inference using such probabilistic grouping over time leads to more robust reasoning, in particular on complex group scenarios. Table 8 provides an overview of group scenarios

Table 8: Probabilistic group-level scenario recognition.

Group scenario	**Probabilities for track i, or between tracks (i,j)**
Group formation	$p_g^f(i) = \text{sigmoid}(y_g^f, 1, 0.2),\ y_g^f = \sum_{\forall j \neq i} p_c^\pi(i,j;t) \cdot \left[1 - p_c^\pi(i,j;t_p)\right] \cdot \max(h(i), h(j))$
Group dispersion	$p_g^d(i) = \text{sigmoid}(y_g^d, 1, 0.2),\ y_g^d = \sum_{\forall j \neq i} p_c^\pi(i,j;t_p) \cdot \left[1 - p_c^\pi(i,j;t)\right] \cdot \max(h(i), h(j))$
Stable group	$p_g^s(i) = 1 - p_g^f(i) - p_g^d(i)$
Loitering group	$p_g^l(i) = 1 - \prod_{\forall j} \left\{1 - p_c^\pi(i,j) p^l(j)\right\}$
Stable loitering group	$p_g^{sl}(i) = p_g^s(i) p_g^l(i)$
Distinct groups	$p_g^\delta(i,j) = \prod_{\forall k} \{1 - \max(p_c^\pi(i,k)p_c^\pi(k,j), p_c^\pi(j,k)p_c^\pi(k,i))\}$
Close-by groups	$p_g^c(i,j;t) = 1 - \sum_{k \neq i,j} \left[1 - p^c(i,k;t)\right] \cdot \left[1 - p^c(k,j;t)\right]$
Group meeting	$p_g^{meet}(i,j) = 1 - \prod_{t=t_0 \text{ to } t_f} \left\{1 - p_g^c(i,j;t)\right\}$
Group following	$p_g^{flw}(i,j) = p_g^\delta(i,j) \cdot \left[1 - \prod_k \left\{p_g^\delta(i,k) + [1 - p_g^\delta(i,k)] \cdot [1 - p^{flw}(k,j)]\right\}\right]$
Group chasing	$p_g^{chs}(i,j) = p_g^\delta(i,j) \cdot \left[1 - \prod_k \left\{p_g^\delta(i,k) + [1 - p_g^\delta(i,k)] \cdot [1 - p^{chs}(k,j)]\right\}\right]$

recognized by our system.

Group structure analysis: We analyze both the static group structures (size, compactness) and their dynamic changes (such as formation, dispersion) over time. The *size* of a group that person i belongs to is estimated as the expected value of the number of *healthy* tracks j that i is connected with:

$$G_s(i) = \sum_{\forall j} p_c^\pi(i,j) h(j), \tag{90}$$

where $h(j)$ is the track healthiness incorporated to deal with false and miss detections, by considering Kalman filter covariance and track lifetime.

We consider three status of a group structure: *(i)* the group is growing (formation), *(ii)* shrinking (dispersion), and *(iii)* remaining the same size (stable), with equations given in Table 8. The idea is to check all the neighbors of person i in $\mathbf{G}$ and see if there is a change in the connectivity. For example, if $\forall_{j \neq i}$, the group connectivity $p_c^\pi(i,j)$ is high at current time t and low at some previous time $t_p = t - T_w$, the probability of group formation of person i is high. We use a time window of $T_w = 30$ frames.

Group scenario recognition: In security, group loitering is of particular interest to municipal-

ities, because it is likely related to (or often the prologue of) illegal activities *e.g.* gang activities and disorderly youth. Our analytical definition of a loitering person has three criteria: *(i)* is currently moving slowly, *(ii)* has been close to the current position at a point in time in the past that was at least T_{min} seconds ago and at most T_{max} seconds ago, and *(iii)* was also moving slowly at that previous point in time.

For each person i, the probability of the belonging group G_i is loitering is one minus the probability that all other individuals in the group are not loitering. This *inversion technique* will be used frequently in subsequent group scenario analysis. An attractive characteristic of our framework is its flexibility to recognize new scenarios by combining existing knowledge. As an example, we detect a *stable loitering group* by multiplying the probability of stable group and group loitering (Table 8). Throughout the paper we denote subscript g as group level probabilities. [4]

Pairwise group structure analysis: We consider two basic types of structure between groups: close-by groups and distinct groups. The former only considers pairwise group relationship at one time step while the latter considers track history. Two groups of a pair of people i and j are considered currently close-by if there exists no person k that both i and j are close to. Using the same inversion technique, we define the probability of close-by groups as: $p_g^c(i,j;t) = 1 - \forall_{k\neq i,j} p\big(k \text{ far from } i \text{ and } k \text{ far from } j \text{ at time } t\big)$.

The distinct groups p_g^δ between individuals i and j is modeled as $\{\forall_k$, there is no connectivity from (i,k) or $(k,j)\}$, using the same inversion technique. In addition, higher-level scenarios such as *stable distinct groups* and *stable loitering distinct groups* can respectively be defined as multiplications of component probabilities:

$$p_g^{s\delta}(i,j) = p_g^s(i)p_g^s(j)p_g^\delta(i,j), \tag{91}$$

$$p_g^{sl\delta}(i,j) = p_g^s(i)p_g^l(i)p_g^s(j)p_g^l(j)p_g^\delta(i,j). \tag{92}$$

Pairwise group scenario recognition: Building upon various per-track basis motion analysis

[4] Although intuitively a group satisfying loitering should be stable in a larger time scale, each individual of it could still be considered not in a stable group in a smaller time scale. We multiply factors by assuming these are independent.

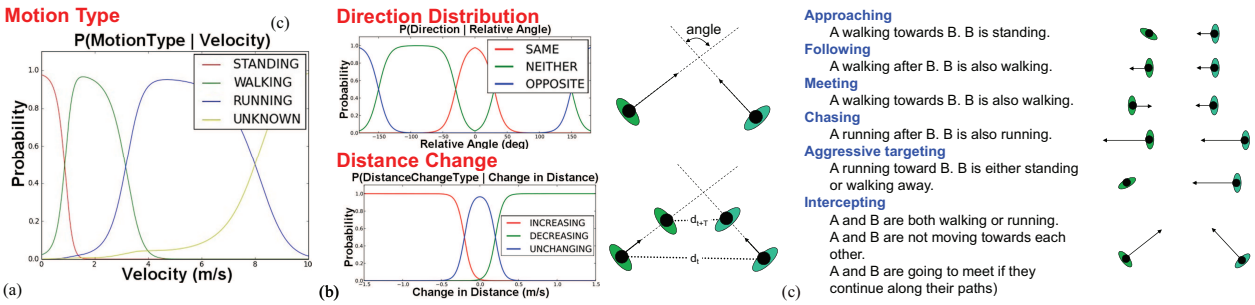


Figure 92: (a) **motion type:** The probability of a person being in different motion types is modeled by a set of sigmoid functions using velocity. (b, top) **relative motion direction:** The probability of different relative moving directions between two people is modeled by a set of sigmoid functions using relative angles. (b, bottom) **relative distance change:** The probability of different relative distance changes between two people is modeled by a set of sigmoid functions using the change in inter-person distance. (c) **pairwise interaction:** Illustration of different types of interaction between two people.

for individuals such as meeting, following, and chasing, we can again recognize group-level scenarios using the soft group representation. The probability of the two groups of a pair of people i and j meeting at time t in the future is defined as:

$$p_g^{meet}(i,j) = 1 - \prod_{t=t_0 \text{ to } t_f} \{1 - p_g^c(i,j;t)\}, \tag{93}$$

where t_0 is the current time, t_f is the time extent in the future, and $p_g^c(i,j;t)$ is the probability of close-by groups.

We define the probability of a group G_i (where person i belongs to) follows an individual j as:

$$p_{gi}^{flw}(i,j) = p_g^{\delta}(i,j) \cdot (1 - p_{nf}(i,j)), \tag{94}$$

where p_{nf}, the probability of G_i not following individual j is defined as

$$p_{nf}(i,j) = \prod_k \big(p_g^{\delta}(i,k) + (1 - p_g^{\delta}(i,k)) \cdot (1 - p^{flw}(k,j))\big).$$

The intuition is that we consider each individual independently, taking account of two cases: either individual k and follower i are in different groups, or the k and i are in one group but k is not following j.

Next we use Eq. 94 to model the case where a group of individuals is following another group:

$$p_g^{flw}(i,j) = p_g^{\delta}(i,j) \cdot (1 - p'_{nf}(i,j)), \tag{95}$$

where the probability of G_i not following G_j is defined similarly as before, by taking account of two cases: either individual k and j are in different groups, or k and j are in the same group but k is not followed by i:

$$p'_{nf}(i,j) = \prod_k \big(p_g^{\delta}(j,k) + (1 - p_g^{\delta}(j,k)) \cdot (1 - p_{gi}^{flw}(i,k))\big).$$

The group-level chasing scenario can be defined similarly, in that the probability p^{chs} of chasing individual should be used. We further define a family of related group-level scenarios such as *group approaching*, *group aggressive targeting*, and *group intercepting*, respective to the pairwise interaction outlined in Fig.92(c) in a similar fashion.

E.5 Probabilistic Individual Motion and Interaction Analysis

This section describes our per-track basis motion analysis, where the equations are summarized in Table 9. We organize the motion scenarios into three main categories (§E.5.1), which serves as the building block for group motion analysis: *(i)* individual motion types, *(ii)* relative motion direction between pairs, and *(iii)* relative distance change between pairs. These components can be combined with the motion prediction modules (Fig.92) described in §E.5.2 to predict several scenarios such as *approaching*, *following*, and *chasing* in §E.5.3.

E.5.1 Person Motion Analysis

We consider the following motion types M^T of a person: *standing*, *walking*, *running*, and an additional *unknown* state. Denote v the velocity of a person, we use the sigmoid function to model the posterior distribution $p(M^T|v) \simeq p(v|M^T)p(M^T)$, visualized in Fig.92(a). We also use velocity measurements to estimate the relative motion direction between pairs of people. We

Table 9: Probabilistic scenario recognition for/between individuals.

Track analysis	Probabilities for track i, or between tracks (i, j)
Track healthiness	$h(i)$, obtained from Kalman filter tracking confidence and lifetime
Person loitering	$p^l(i)$ (omitted in this paper due to space limit)
Person motion type	$p^{mt}(M^T\|, i)$, $M^T = \{$ *standing, walking, running, unknown* $\}$, Fig.92(a)
Relative motion direction	$p^{md}(M^D\|i, j)$, $M^D = \{$ *same, opposite, neither* $\}$, Fig.92(b, upper)
Relative distance change	$p^{dc}(D^C\|i, j)$, $D^C = \{$ *increasing, decreasing, unchanging* $\}$, Fig.92(b, lower)
Track to track pairwise metric	$p_c^p(i, j)$ incorporating front/sided-ness, velocity, and motion track history in Eq.87
Track to track path connectivity	$p_c^\pi(i, j)$, obtained from $p_c^p(i, j)$ after all-pair shortest path computation
Person meeting	$p^{meet}(i, j) = 1 - \prod_{t=t_0 \text{ to } t_f} \{1 - p^c(i, j; t)\}$, where p^c is defined in Eq.96
Person following	$p^{flw}(i, j) = p(M_i^T = \textit{walking})p(M_j^T = \textit{walking})\left[1 - \text{sigmoid}(d_{itc}(i, j), \mu_{d_{itc}}, \sigma_{d_{itc}})\right]$
Person chasing	$p^{chs}(i, j) = p(M_i^T = \textit{running})p(M_j^T = \textit{running})\left[1 - \text{sigmoid}(d_{itc}(i, j), \mu_{d_{itc}}, \sigma_{d_{itc}})\right]$

allow three relative motion directions $M^D = \{same, opposite, neither\}$, which are conditioned on the angle between their velocity vectors. A graphical visualization of the posterior is shown in Fig.92(b, upper).

E.5.2 Motion Prediction

We utilize the time varying prediction of location probability distributions produced by the Kalman tracker to reason about the chance if two people will spatiotemporally get close. This probability is used in group-level recognition such as close-by and the meeting of groups.

Denote the location prediction of individuals i and j as $\mathbf{x}^i(t) \simeq N(\mathbf{z}_t^i, \mathbf{S}_t^i)$ and $\mathbf{x}^j(t) \simeq N(\mathbf{z}_t^j, \mathbf{S}_t^j)$ respectively, where N denotes normal distribution, $\mathbf{z}_t$ is the estimated ground plane location and $\mathbf{S}_t$ is the associated uncertainty. The probability that two people are close to each other at time t can be estimated by assuming that the true locations of i and j are indeed $\mathbf{x}^i(t)$ and $\mathbf{x}^j(t)$: $p^c(i, j; t) = \theta\ (||\mathbf{x}^i(t) - \mathbf{x}^j(t)|| - \sigma_c)$, where θ denotes thresholding. However, we do not know the exact location of i and j at time t. We hence perform a numerical integration over all possible locations with a set of sample points and weights that represent the distributions $N(\mathbf{x}^i; \mathbf{z}_t^i, \mathbf{S}_t^i)$ and $N(\mathbf{x}^j; \mathbf{z}_t^j, \mathbf{S}_t^j)$. The predicted probability of two people to be close at time t is:

$$p^c(i, j; t|\mathbf{z}_t^i, \mathbf{S}_t^i, \mathbf{z}_t^j, \mathbf{S}_t^j) = \sum_{m,n} (\theta(||\mathbf{x}_m^i - \mathbf{x}_m^j|| - \sigma_c) w_m^i w_m^j). \tag{96}$$

E.5.3 Recognizing Pairwise Interaction Scenarios

In estimating probabilistic *meeting*, *i.e.* to determine if two individuals will meet in a future time interval $t \in [0, T]$, it is not trivial to remove the time dependency by marginalizing over the time t, since locations $\mathbf{x}^i(t)$ and $\mathbf{x}^j(t)$ are not necessarily independent between time steps. We hence chose to select discrete time slices t_a and infer how probable it is that two targets are going to be close at least at one time t_s from a set $\{t_s | s = 0, \ldots, N-1\}$. This is again the inversion technique that $p^{meet}(i, j)$ equals 1 minus the probability that the targets *never* got close (Table 9).

Probabilistic *following/chasing* is estimated based on the person motion types and the prediction of future locations. In a following event, we first compute the interception distance $d_{itc}(i, j)$ between a follower i and a target j by computing the shortest distance between the current group location of j and the predicted location of i (extrapolated using the current velocity estimation). If both people are walking and their interception distance is small, we consider it a following event. Similarly, for both are running, it is then a chasing event (Table 9).

Other scenarios such as probabilistic *approaching*, *aggressive targeting*, and *intercepting*, summarized in Fig.92(c) can be modeled similarly by incorporating motion pattern and motion prediction modules. These can be combined to detect high-level scenarios. For example, the probability of two targets quickly running toward, and meeting at the same location is:

$$\begin{aligned}\left[p^{md}(\textit{opposite}|i,j) + p^{md}(\textit{neither}|i,j)\right] \cdot p^{dc}(\textit{decreasing}|i,j) \\ \cdot p^{mt}(\textit{running}|i) \cdot p^{mt}(\textit{running}|j) \cdot p^{meet}(i,j).\end{aligned}$$

E.6 Implementation, Results and Evaluation

Video tracking system: Our system is equipped with four standard CCTV cameras for data collection and testing. The tracking system [38] comprises of multiple calibrated and synchronized cameras performing tracking cooperatively. Person detections in each view are projected onto the ground plane in 3D and are then fed into a centralized Kalman tracker operating on the ground

Figure 93: Snapshots of several group-level activities captured by our system, where in each case the top depicts one (out of many) camera views and the bottom depicts a top-down 2D planar view, except (h). See text for details.

plane. Our system runs automatically in real-time (15 to 30 fps) in recognizing pre-defined events for about 5-20 people or more.

To ensure real-life performance, was first deployed our system in a courtyard to test on pedestrian surveillance. We then participate an official field test on the MPR dataset (http://mockprisonriot.org), where several correction officers volunteered to enact security relevant behaviors such as agitated arguments, fights, contraband exchange in an abandoned prison yard in West Virginia, USA. As many activities of interest for correctional settings are related to gang activities, the enacted scenarios often has the presence of multiple groups.

Fig.93 illustrates snapshots of several scenarios detected by our system. Fig.93(a) shows our soft grouping connectivity in the prison yard scenario when 6 enacted inmates are about to form

two groups to challenge each other and fight. Our path-based probabilistic grouping scheme successfully identifies the two main groups. Fig.93(b-h) depicts samples of detected group activities in several scenarios selected from both datasets. Specifically, in Fig.93(a), cyan lines depict our probabilistic path-based grouping connectivity p_c^{π} as a complete graph of all tracks; lower transparency indicates higher probability. Fig.93(b) shows detected stable loitering groups, where the transparency indicates probability p_g^{sl}. Fig.93(c) shows three detected pairs of stable distinct groups, where each yellow edge depicts $p_g^{s\delta}$. Observe that only people of different groups trigger strong signals. Fig.93(d) shows detected group formation, where the 4 people gather together such that their overall group size is increasing. We denote 'F' in red for formation, 'S' in green for stable, and their transparency indicates probabilities p_g^f and p_g^s. Fig.93(e) shows detected group dispersion in the same sequence after (d), where the 4 individuals disperse ('D' in yellow for p_g^d). Fig.93(f) shows detected group following p_g^{flw} in red edges connecting two major groups, where one is following another. Fig.93(g) shows detected flanking maneuver, where 4 enacted inmates are surrounding 2 victims for an attack. Red circle indicates high probability of being flanked, while green circle indicates low probability. Observe that the grouping connectivity between members in the aggressive group is very strong. Fig.93(h) shows detected contraband exchange with a certain probability.

We next discuss two specific scenarios in related security applications, where the recognition involves complex spatial-temporal reasoning over low-level and high-level interpretations. We show that our probability framework is flexible and adaptable to solve them.

Case study I: The scenario of *flanking maneuver* is a spatiotemporal configurations exhibited by groups, where one or more aggressive and dominating groups spread out to surround a victim group prior to an attack. Refer to Fig.93(g) for an example occurred in a prison yard. We consider the probabilistic flanking condition $p^{flk}(i, j : k)$, where a victim k is flanked by two others i and j if: (1) i, j are in a group, i, k and j, k are in different groups, the distance $d(i, j)$ is large enough than $d(i, k)$ and $d(j, k)$, and the angle θ_f between $\overrightarrow{ki}$ and $\overrightarrow{kj}$ is large enough. The event probability is then:

$$p^{flk}(i,j:k) = p_c^{\pi}(i,j)\left[1-p_c^{\pi}(i,k)\right]\left[1-p_c^{\pi}(j,k)\right]\cdot \tag{97}$$
$$\text{sigmoid}(d_{ratio},\mu_{d_r},\sigma_{d_r})\cdot\text{sigmoid}(\theta,\mu_\theta,\sigma_\theta),$$

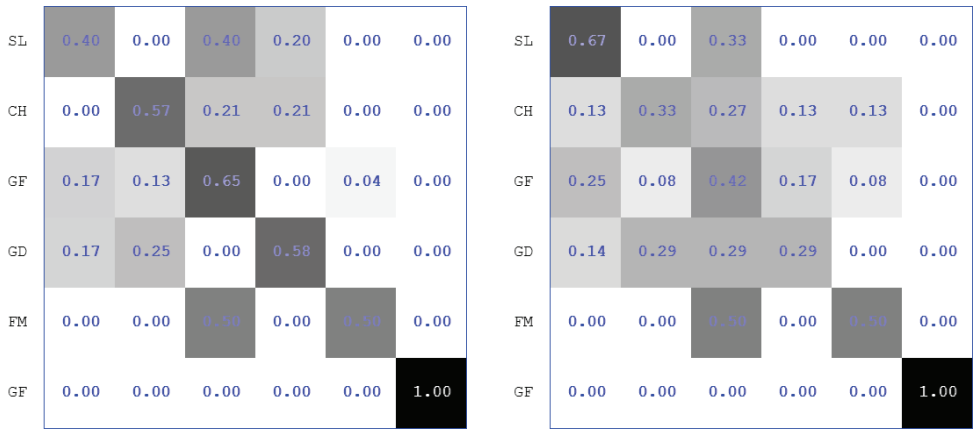
where μ_θ and σ_θ control how wide should the attackers i and j spread in order to flank the victim k; the distance radio $d_{ratio} = \frac{2d(i,j)}{d(i,k)+d(j,k)}$ controls the proper distance between the attackers and the victim. We consider all pairs of i and j for every individual k in accumulating all probabilities. Thus, the probability of individual k is flanked is $1 - p(k \text{ is not flanked by any others})$:

$$p^{flk}(k) = 1 - \prod_{\forall \text{ pairs of } i,j}\left[1-p^{flk}(i,j:k)\right]. \tag{98}$$

Case study II: Another important application in prison security is to monitor if there is any *contraband handoff* between inmates, where improvised knives, drugs, messages and others are exchanged. This scenario requires two individuals to physically get close to each other during the handoff, Fig.93(h) and also that a short while of T seconds ago, the individuals are not close yet but are approaching each other. There is at least one person walking during the approach, while the second person might be standing or walking. The scenario is modeled as $p(\{\text{i and j are exchanging contraband at time t}\}) =$

$$[p^{mt}(\textit{walking}|i;t_p = t-T) + p^{mt}(\textit{walking}|j;t_p) -$$
$$p^{mt}(\textit{walking}|i;t_p)p^{mt}(\textit{walking}|j;t_p)]\cdot$$
$$p^{meet}(i,j;t_p)p^{md}(\textit{opposite}|i,j,t_p)\left[1-p^{c}(i,j;t_p)\right]p^{c}(i,j;t).$$

Validation: We have performed evaluation on various group-level events against manually labelled ground truth as shown in Fig.94. The confusion matrix labels are in the order of stable loitering group (SL), contraband handoff (CH), group formation (GF), group dispersion (GD), flanking maneuver (FM), and group following (GF). Note that our evaluation uses videos with complex group interactions where multiple events of interest can occur simultaneously. For example, prior



(a) Ground truth (b) Test results

Figure 94: Confusion matrix comparing the accuracy of the group-level event classification (see text).

to a gang fighting event while a group of people is loitering, another group can be forming. The contraband hand-off will inevitably include a meeting as a part. Therefore, the diagonal values are not always 1 in the ideal cases as a direct result of composite events (as compared to a conventional confusion matrix) shown in Fig.94(a). Fig.94(b) shows our test result, which fairly resembles the table generated from the ground truth.

E.7 Conclusion

We have described a probabilistic framework to recognize group-level activity in many scenarios using a novel soft grouping metric and track-based motion analysis. We use a graph between individuals within a scene to determine group membership and interactions, where sound probabilistic estimates can be combined to handle new scenarios. The approach is per-track basis, fully automatic and efficient. In the future, for crowded scenes we can augment our method by first generating a large scale clustering and for each cluster perform detailed group analysis. Another future direction is to recover minor tracking errors by exploiting the probabilistic reasoning about the groups as a feedback.

Acknowledgement. This work was supported by grant #2009-SQ-B9-K013 awarded by the National Institute of Justice. The opinions, findings, and conclusions or recommendations ex-

pressed in this publication are those of the authors and do not necessarily reflect the views of the Department of Justice.

F Advanced Gaze Tracking

The following is a reprint of a manuscript that was presented and published at the IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS) 2011. The precise title of the paper is: Ming-Ching Chang, Nils Krahnstoever, and Weina Ge, "Gaze and Body Pose Estimation from a Distance", IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), Klagenfurt, Austria, Aug. 2011. This paper won the Best Paper (Runner Up) Award at the conference.

Abstract

We present a comprehensive approach to track gaze by estimating location, body pose, and head pose direction of multiple individuals in unconstrained environments. The approach combines person detections from fixed cameras with directional face detections obtained from actively controlled pan tilt zoom (PTZ) cameras. The main contribution of this work is to estimate both body pose and head pose (gaze) direction independently from motion direction, using a combination of sequential Monte Carlo Filtering and MCMC sampling. There are numerous benefits in tracking body pose and gaze in surveillance. It allows to track people's focus of attention, can optimize the control of active cameras for biometric face capture, and can provide better interaction metrics between pairs of people. The availability of gaze and face detection information also improves localization and data association for tracking in crowded environments. The performance of the system will be demonstrated on data captured at a real-time surveillance site.

F.1 Introduction

Detecting and tracking individuals under unconstrained conditions such as in mass transit stations, sport venues, and schoolyards are important. On top of that, the understanding of their gaze and intention are more challenging due to the general freedom of movements and frequent occlusions. Moreover, face images in standard surveillance videos are usually low-resolution, which limits the detection rate. Unlike previous approaches [82, 83, 84] that at most obtained gaze information, we

use multi-view pan tilt zoom (PTZ) cameras and tackle the problem of joint, holistic tracking of both body pose and head orientation in real-time. Following Stiefelhagen *et al.* [85], we assume that the gaze can be reasonably derived by head pose in most cases [82]. Throughtout the paper we refer to head pose as gaze or visual focus of attention and use them interchangably. The coupled person tracker, pose tracker, and gaze tracker are integrated and synchronized, thus robust tracking via mutual update and feedback is possible. The capability to reason over gaze angle provides a strong indication of attention, which benifits a surveillance system on many fronts. In particular as part of interaction models in event recognition, it is important to know if a group of individuals are facing each other (*e.g.*, talking), facing a common direction (*e.g.*, looking at another group before a conflict is about to happen) or facing away from each other (*e.g.*, because they are not related or because they are in a "defense" formation).

Our contribution is three-fold. *(1)* We propose a unified framework to couple multi-view person tracking with asynchronous PTZ gaze tracking to jointly and robustly estimate pose and gaze. The novelty over [86] is a coupled particle filtering tracker that jointly estimates body pose and gaze. While we use person tracking to control PTZ cameras, which allow us to perform face detection and gaze estimation, we in turn utilize the resulting face detection locations to further improve tracking performance. *(2)* We can thus actively leverage track information to control PTZ cameras in maximizing the probability of capturing frontal facial views. Our work significantly improves previous efforts [87] that used the walking direction of individuals as an indication of gaze direction, which breaks down in situations where people are stationary. *(3)* Our framework is general and applicable to many other vision-based applications. For example, we allow optimal face capture for biometrics particularly in environments where people are stationary, because it obtains gaze information directly from face detections.

We use a network of fixed cameras to perform site-wide person tracking. This person tracker drives one or more PTZ cameras to target individuals (details in §F.3) to obtain close-up views. A centralized tracker operates on the ground plane to fuse together information from person tracks and face tracks. Due to the large computational burden on inferring gaze from face detections, the

person tracker and face tracker must operate *asynchronously* to run in real-time. Our system can operate on either a single or multiple cameras. The multi-camera setting improves overall tracking performance in crowded conditions. Gaze tracking in this case is also useful in performing high-level reasoning *e.g.*, to analyze social interactions [88], attention model, and behaviors [89].

F.2 Related Work

To the best of our knowledge, we are the first to proactively integrate multi-camera with multi-PTZ pose estimation for gaze tracking in unconstrained environments. Our work involves multi-view person tracking and head pose tracking across one or more views. In practice the face resolution is typically low, so one must either rely on special methods [90, 84] or use PTZ camera [86] to obtain close-up shots (as we did). The method of Robertson *et al.* [84] is data-driven based on Bayesian fusion of priors, thus relies on training videos. Their coupling of face angle and head tracking workes in a limited single field of view. Hoedl *et al.* [91] use a two-camera system (one fixed and one PTZ) to perform pedestrian detection, however no face or gaze analysis is performed.

Head pose and gaze tracking has been studied extensively [53]. However, most existing systems restrict to an indoor room setting and it is often assumed that subjects stay seated (therefore tracking is trivial and no camera control is involved).

The idea of joint person and face tracking is not new, however, existing works do not attempt to fuse both trackers in using one to update the other. Ozturk *et al.* [92] track body and head pose using independent trackers on a single top-view camera. Bäuml *et al.* [57] assume head location is known and track faces across a distributed camera network for recognition and re-identification. Their face tracker and person tracker runs separately, where the overlapping facial views are processed independently. Smith *et al.* [82] estimate multiple individuals' body pose/gaze to track their visual focus of attention. Their work is relevant to us, except that they use a single camera view, while we fuse the tracking across multiple views.

F.3 Video Tracking and PTZ Control

Our video tracking system is based on [93] consisting of 4 fixed and 4 PTZ cameras. The fixed camera views are utilized by a centralized tracker to estimate the 3D locations of individuals in a common ground plane. The person tracking are then used online to drive the PTZ cameras to capture zoom-in face images. Recognized faces are then associated with person trackers to cooperatively improve the overall tracking.

Our PTZ camera control strategy aims at capturing frontal face views from individuals, even including uncooperative ones at a distance. The system must then determine how (what camera to drive, and where to move) to obtain the best shots automatically. Our control algorithm pursues the goal of scheduling PTZ cameras in in a way optimizing frontal face capture. The control system provides each PTZ camera with a continuously evolving *schedule* [87] that describes what targets to visit in what order. Schedules are planned several target capture steps into the future based on the current and predicted motion of observed individuals. A given schedule is assigned a probability of achieving the goal of continuously capturing high quality facial shots of all tracked individuals. The quality of facial shots is governed by several factors: the distance of individuals from the camera, the angle at which a face is captured, and the accuracy with which a person is being located by the tracker. A control strategy is chosen by selecting the schedule with the highest probability from the set of all possible schedules. Our PTZ control is pursuing a chicken and egg problem. It leverages gaze information to schedule the PTZ cameras, but (at least for stationary individuals) no gaze information will be acquired until close-up PTZ views have been obtained. The controller hence builds the uncertainty around the gaze estimates into the control strategy.

F.4 Person, Body Pose and Gaze Tracking

F.4.1 Problem Definition

We represent each individual with a state vector $\mathbf{s} = [\mathbf{x}, \mathbf{v}, \alpha, \phi, \theta]$, where $\mathbf{x}$ is the location on the (X,Y) groundplane metric world, $\mathbf{v}$ is the velocity on the groundplane, α is the horizontal

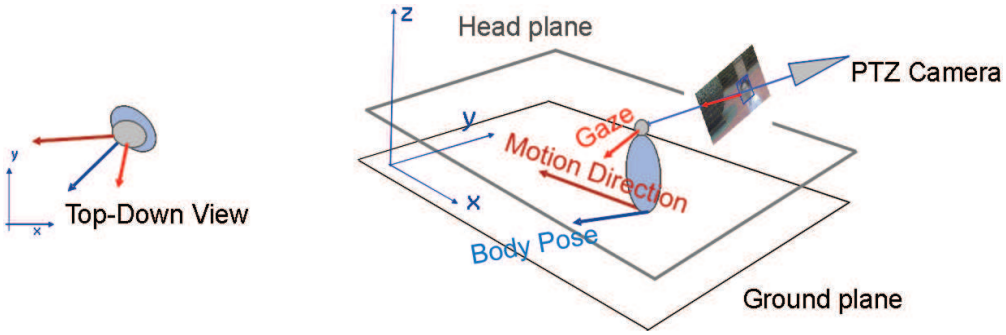


Figure 95: We are estimating each person's location ($\mathbf{x}$), velocity ($\mathbf{v}$), body pose ($\boldsymbol{\alpha}$) and gaze direction ($\boldsymbol{\phi}, \boldsymbol{\theta}$) in world coordinates.

orientation of the body around the groundplane normal, ϕ is the horizontal gaze angle and θ is the vertical gaze angle (positive above the horizon and negative below it), see Fig.95. There are two types of observations in our system: person detections $(\mathbf{z}, \mathbf{R})$, where $\mathbf{z}$ is a groundplane location measurement and $\mathbf{R}$ the uncertainty of this measurement and face detections $(\mathbf{z}, \mathbf{R}, \gamma, \rho)$ where the additional parameters γ and ρ are the horizontal and vertical gaze angles. Each person's head and foot locations are extracted from image-based person detections [93] and back-projected onto the world head- and ground-plane respectively, using an unscented transform (UT). Next, face positions and poses in PTZ views are obtained using the PittPatt face detector [64]. Their metric world groundplane locations are again obtained through back-projection. Face pose is obtained by matching face features. Individual's gaze angles are obtained by mapping face pan and rotation angles in image space into the world space. Finally, the world gaze angles are obtained by mapping the image local face normal $\mathbf{n}_{\text{img}}$ into world coordinates via $\mathbf{n}_{\text{w}} = \mathbf{n}_{\text{img}}\mathbf{R}^{-T}$, where $\mathbf{R}$ is the rotation matrix of the projection $\mathbf{P} = [\mathbf{R}|\mathbf{t}]$. Observation gaze angles (γ, ρ) are obtained directly from this normal vector. Width and height of the face are used to estimate a covariance confidence level for the face location. The covariance is projected from the image to the groundplane again using the UT from the image to the head plane, followed by down projection to the groundplane. Fig.97 depicts detected faces and gaze angles in estimation in multiple views.

In contrast to [86], where a person's gaze angle was estimated independently from location and velocity, and body pose was ignored, this work aims at correctly modeling the relationship between motion direction, body pose, and gaze. First, in this work body pose is not strictly tied to

motion direction. People can move backwards and sideways especially when people are waiting or standing in groups (*albeit*, with increasing velocity sideways people's motion becomes improbable, and at even greater velocities, only forward motion is assumed). Secondly, head pose is not tied to motion direction, but there are relatively strict limits on what pose the head can assume relative to body pose. Under this model the estimation of body pose is not trivial as it is only *loosely coupled* to gaze angle and velocity (which in turn is only observed indirectly). We perform the entire state estimation using a Sequential Monte Carlo filter, described in the next section.

F.4.2 Estimation of Location, Pose and Gaze

We assume for now that we have a method for associating measurements with tracks over time. For the sequential Monte Carlo filter, we need to specify *(i)* the dynamical model and *(ii)* the observation model of our system.

Dynamical Model: Following the description in the previous section, our state vector is $\mathbf{s} = [\mathbf{x}, \mathbf{v}, \alpha, \phi, \theta]$ and the state prediction model decomposes as follows:

$$\begin{aligned} p(\mathbf{s}_{t+1}|\mathbf{s}_t) = p(\mathbf{q}_{t+1}|\mathbf{q}_t)p(\alpha_{t+1}|\mathbf{v}_{t+1}, \alpha_t) \\ p(\phi_{t+1}|\phi_t, \alpha_{t+1})p(\theta_{t+1}|\theta_t), \end{aligned} \tag{99}$$

where we used the abbreviation $\mathbf{q} = (\mathbf{x}, \mathbf{v}) = (x, y, v_x, v_y)$. For the location and velocity we assume a standard linear dynamical model

$$p(\mathbf{q}_{t+1}|\mathbf{q}_t) = \mathcal{N}(\mathbf{q}_{t+1} - \mathbf{F}_t\mathbf{q}_t, \mathbf{Q}_t), \tag{100}$$

where $\mathcal{N}$ denotes Normal distribution, $\mathbf{F}_t$ is a standard constant velocity state predictor corresponding to $\mathbf{x}_{t+1} = \mathbf{x}_t + \mathbf{v}_t\Delta t$ and $\mathbf{Q}_t$ the standard system dynamics [94]. The second term in Eq.(99) describes the propagation of the body pose under consideration of the current velocity

vector. We assume the following model

$$p(\alpha_{t+1}|\mathbf{v}_{t+1}\alpha_t) = \mathcal{N}(\alpha_{t+1} - \alpha_t, \sigma_\alpha) \cdot \tag{101}$$
$$\begin{cases} (1.0 - P^o)\mathcal{N}(\alpha_{t+1} - \nu_{t+1}, \sigma_{\nu\alpha}) + P^o \frac{1}{2\pi} & \text{if } ||\mathbf{v}|| > 2 \text{ m/s}, \\ \frac{1}{2\pi} & \text{or if } ||\mathbf{v}|| < \frac{1}{2} \text{ m/s}, \\ P^f \mathcal{N}(\alpha_{t+1} - \nu_{t+1}, \sigma_{\nu\alpha}) + & \\ \quad P^b \mathcal{N}(\alpha_{t+1} - \nu_{t+1} - \pi, \sigma_{\nu\alpha}) + P^o \frac{1}{2\pi} & \text{otherwise}, \end{cases}$$

where $P^f = 0.8$ is the probability (for medium velocities $\frac{1}{2}$ m/s $< \mathbf{v} <$ 2 m/s) of a person walking forwards, $P^b = 0.15$ the probability (for medium velocities) of walking backwards, and $P^o = 0.05$ the background probability allowing arbitrary pose to movement direction relationships, based on experimental heuristics. With ν_{t+1} we denote the direction of the velocity vector $\mathbf{v}_{t+1}$ and with $\sigma_{\nu\alpha}$ the expected distribution of deviations between movement vector and body pose. The front term $N(\alpha_{t+1} - \alpha_t, \sigma_\alpha)$ represents the system noise component, which in turn limits the change in body pose over time. All changes in pose are attributed to deviations from the constant pose model.

The third term in Eq.(99) describes the propagation of the horizontal gaze angle under consideration of the current body pose. We assume the following model

$$\begin{aligned} &p(\phi_{t+1}|\phi_t\alpha_{t+1}) = \mathcal{N}(\phi_{t+1} - \phi_t, \sigma_\phi) \cdot \\ &\left\{ P_g^u \Theta(|\phi_{t+1} - \frac{\pi}{3}|) + P_g \mathcal{N}(\phi_{t+1} - \alpha_{t+1}, \sigma_{\alpha\phi}) \right\}, \end{aligned} \tag{102}$$

where the two terms weighted by $P_g^u = 0.4$ and $P_g = 0.6$ define a distribution of the gaze angle (ϕ_{t+1}) with respect to body pose (α_{t+1}) that allows arbitrary values within a range of $\alpha_{t+1} \pm \frac{\pi}{3}$ but favors distribution around body pose.

Finally the fourth term in Eq.(99) describes the propagation of the tilt angle, $p(\theta_{t+1}|\theta_t) = \mathcal{N}(\theta_{t+1}, \sigma_\theta^0)\mathcal{N}(\theta_{t+1} - \theta_t, \sigma_\theta)$, where the first term models that a person tends to favor horizontal directions and the second term represents system noise. Noted that in all above equations, care has

to be taken with regard to angular differences.

To propagate the particles forward in time, we need to sample from the state transition density Eq.(99), given a previous set of weighted samples $(\mathbf{s}_t^i, \mathbf{w}_t^i)$. While for the location, velocity and vertical head pose, this is easy to do. The loose coupling between velocity, body pose and horizontal head pose is represented by a non-trivial set of transition densities Eq.(101) and Eq.(102). To generate samples from these transition densities we perform two Markov Chain Monte Carlo (MCMC). Exemplified on Eq.(101), we use a Metropolis sampler [95] to obtain a new sample as follows:

- **Start:** Set $\alpha_{t+1}^i[0]$ to be the α_t^i of particle i.
- **Proposal Step:** Propose a new sample $\alpha_{t+1}^i[k+1]$ by sampling from a *jump-distribution* $G(\alpha|\alpha_{t+1}^i[k])$.
- **Acceptance Step:** Set $r = p(\alpha_{t+1}^i[k+1]|\mathbf{v}_{t+1}\alpha_t^i)/p(\alpha_{t+1}^i[k]|\mathbf{v}_{t+1}\alpha_t^i)$. If $r \geq 1$, accept the new sample. Otherwise accept it with probability r. If it is not accepted, set $\alpha_{t+1}^i[k+1] = \alpha_{t+1}^i[k]$.
- **Repeat:** Until $k = N$ steps have been completed.

Typically only a small fixed number of steps ($N = 20$) are performed. The above sampling is repeated for the horizontal head angle in Eq.(102). In both cases the jump distribution is set equal to the system noise distribution, except with a fraction of the variance *i.e.*, $G(\alpha|\alpha_{t+1}^i[k]) = \mathcal{N}(\alpha - \alpha_{t+1}^i[k], \sigma_\alpha/3)$ for body pose; $G(\phi|\phi_{t+1}^i[k])$ and $G(\theta|\theta_{t+1}^i[k])$ are defined similarly. The above MCMC sampling ensures that only particles that adhere both to the expected system noise distribution as well to the loose relative pose constraints are generated. We found 1000 particles are sufficient.

Observation Model: After sampling the particle distribution $(\mathbf{s}_t^i, \mathbf{w}_t^i)$ according to its weights $\{\mathbf{w}_t^i\}$ and forward propagation in time (using MCMC as described above), we obtain a set of new samples $\{\mathbf{s}_{t+1}^i\}$. The samples are weighted according to the observation likelihood models described next.

Algorithm 1: **Location, pose and gaze angle tracking.**

Data : Sample set $S_t = (w_t^i, \mathbf{s}_t^i)$

Result : Sample set $S_{t+1} = (w_{t+1}^i, \mathbf{s}_{t+1}^i)$

begin

  for $i = 1, \ldots, M$ *(number of particles)* **do**

    Randomly select sample $\mathbf{s}_t^i = (\mathbf{x}_t^i, \mathbf{v}_t^i, \alpha_t^i, \phi_t^i, \theta_t^i)$ from S_t according to weights w_t^i

    Obtain forward propagated locations $\mathbf{x}_{t+1}^i$ and $\mathbf{v}_{t+1}^i$ by sampling from distribution Eq.(100).

    Perform MCMC to sample a new body pose α_{t+1}^i from Eq.(101).

    Perform MCMC to sample a new horisontal gaze vector ϕ_{t+1}^i from Eq.(102).

    Sample new vertical face angle θ_{t+1}^i from distribution $p(\theta_{t+1}|\theta_t)$.

    Evaluate new state ${w_{t+1}^i}' = p(\mathbf{Z}_{t+1}|\mathbf{s}_{t+1}^i)$ with Eq.(103) if the observation is a person detection, or Eq.(104) if it is a directional face detection. Renormalize particle set to obtain final update distribution $S_{t+1} = (w_{t+1}^i, \mathbf{s}_{t+1}^i)$.

end

For the case of person detections, the observations are represented by $(\mathbf{z}_{t+1}, \mathbf{R}_{t+1})$ and the likelihood model is:

$$p(\mathbf{z}_{t+1}|\mathbf{s}_{t+1}) = \mathcal{N}(\mathbf{z}_{t+1} - \mathbf{x}_{t+1}|\mathbf{R}_{t+1}). \tag{103}$$

For the case of face detection $(\mathbf{z}_{t+1}, \mathbf{R}_{t+1}, \gamma_{t+1}, \rho_{t+1})$, the observation likelihood model is

$$\begin{aligned} p(\mathbf{z}_{t+1}, \gamma_{t+1}, \rho_{t+1}|\mathbf{s}_{t+1}) &= \mathcal{N}(\mathbf{z}_{t+1} - \mathbf{x}_{t+1}|\mathbf{R}_{t+1}) \\ &\mathcal{N}(\lambda((\gamma_{t+1}, \rho_{t+1}), (\phi_{t+1}, \theta_{t+1})), \sigma_\lambda), \end{aligned} \tag{104}$$

where $\lambda(.)$ is the geodesic distance (expressed in angles) between the points on the unit circle represented by the gaze vector $(\phi_{t+1}, \theta_{t+1})$ and the observed face direction $(\gamma_{t+1}, \rho_{t+1})$ respectively.

$$\begin{aligned} \lambda((\gamma_{t+1}, \rho_{t+1}), (\phi_{t+1}, \theta_{t+1})) = \arccos(&\sin\rho_{t+1}\sin\theta_{t+1} \\ &+ \cos\rho_{t+1}\cos\theta_{t+1}\cos(\gamma_{t+1} - \phi_{t+1})). \end{aligned}$$

The value σ_λ is the uncertainty that is attributed to the face direction measurement. Overall the tracking state update process works as summarized in Algorithm 1.

F.4.3 Data Association

So far we assumed that observations had already been assigned to tracks. In this section we will elaborate how observation to track assignment is performed. To enable the tracking of multiple people, observations have to be assigned to tracks over time. In our system, observations arise *asynchronously* from multiple camera views. The observations are projected into the common world reference frame, under consideration of the (possibly time varying) projection matrices, and are consumed by a centralized tracker in the order that the observations have been acquired. For each time step, a set of (either person or face) detections $\mathbf{Z}_t^l$ have to be assigned to tracks $\mathbf{s}_t^k$. We construct a distance measure $C_{kl} = d(\mathbf{s}_t^k, \mathbf{Z}_t^j)$ to determine the optimal one-to-one assignment of observations l to tracks k using Munkres algorithm [63]. Observations that do not get assigned to tracks might be confirmed as new targets and are used to spawn new candidate tracks. Tracks that do not get detections assigned to them are propagated forward in time and thus do not undergo weight update.

The use of face detections lead to an additional source of location information. We explicitly use this to improve tracking. Results show that this is particularly useful in crowded environments, where face detectors are less susceptible to person-person occlusion. Our other advantage is that the gaze information introduces an additional component into the detection-to-track assignment distance measure, which works effectively to assign oriented faces to person tracks.

For person detections, the metric is computed from the target gate as follows:

$$\boldsymbol{\mu}_t^k = \frac{1}{N}\sum_i \mathbf{x}_t^{ki},\ \boldsymbol{\Sigma}_t^{kl} = \frac{1}{N-1}\sum_i (\mathbf{x}_t^{ki} - \boldsymbol{\mu}_t^k)(\mathbf{x}_t^{ki} - \boldsymbol{\mu}_t^k)^T + \mathbf{R}_t^l,$$

where $\mathbf{R}_t^l$ is the location covariance of observation l and $\mathbf{x}_t^{ki}$ is the location of the i^{th} particle of track k at time t. The distance measure is then given as:

$$C_{kl}^l = (\boldsymbol{\mu}_t^k - \mathbf{z}_t^l)^T(\boldsymbol{\Sigma}_t^{kl})^{-1}(\boldsymbol{\mu}_t^k - \mathbf{z}_t^l) + \log|\boldsymbol{\Sigma}_t^{kl}|$$

For face detections, the above is augmented by an additional term for the angle distance:

$$C_{kl} = C_{kl}^{l} + \frac{\lambda((\gamma_t^l, \rho_t^l), (\mu_{\phi_t}^k, \mu_{\theta_t}^k))^2}{\sigma_\lambda^2} + \log \sigma_\lambda^2,$$

where the $\mu_{\phi_t}^k$ and $\mu_{\theta_t}^k$ are computed from the first order spherical moment of all particle gaze angles (angular mean); σ_λ is the standard deviation from this moment; (γ_t^l, ρ_t^l) are the horizontal and vertical gaze observation angles in observation l.

Since only PTZ cameras provide face detections and only fixed cameras provide person detections, data association is performed with either all person detections or all face detections; the gaze of mixed associations does not arise.

F.5 Experiments and Results

We first demonstrate the concept of our system with a single person, without the use of face detection information. In this mode the estimation reduces to the prior of the body pose and gaze, conditioned on the motion vector of the person. As one can see in Fig.96, for moderate walking speeds the system correctly estimates the direction to be either forward or backward facing, relative to the motion vector (representing the two possibilities of the person walking forward or backward), and the gaze vector is estimated to fall within a range of these forward and backward directions. Figs. 96 to 99 visualize the gaze and body pose estimates via circular histograms around each person, where the fan of circular bars radiate away from each target. The direction of the bars indicate different pose and gaze angles; the length of the bars indicates the probability associated with this direction. In Fig.96 the yellow histogram around the head represent gaze direction and the green histogram around the feet represent bode pose direction. The two green lines radiating from the head are the one sigma standard deviations from the average gaze direction and the red lines radiating from the feet are the one sigma deviations of the body pose. Also shown are the trajectory in the ground plane (dark red). Fig.96(b) shows how for slow backward motion the ambiguity between forward or backward pose increases, represented by the two almost equally

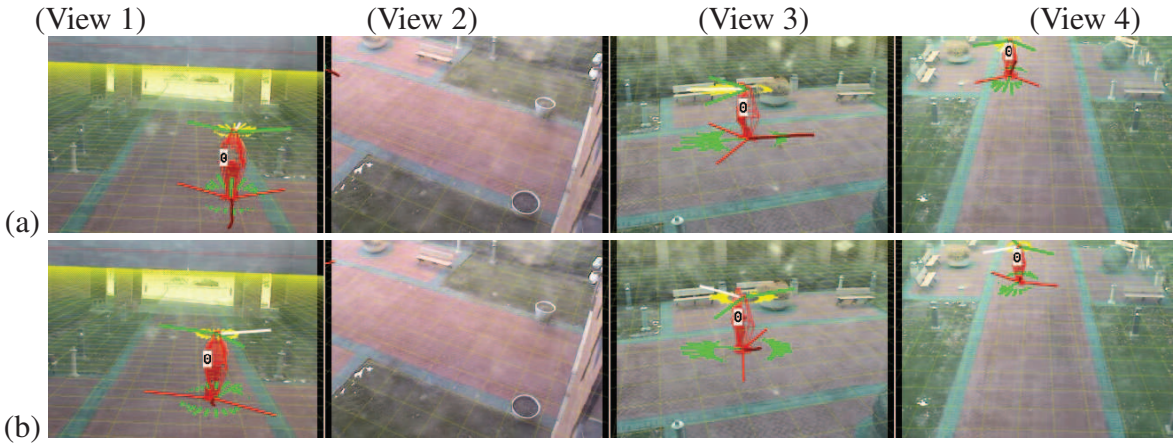


Figure 96: Pose and gaze estimation in absence of face detections: (a) forward motion, (b) backward motion. See text for explanation.

distributed direction possibilities.

We repeated the single person tracking experiment with the face detector enabled in Fig.97. We can clearly see how the introduction of gaze measurements have significantly improved the accuracy of the gaze and pose estimates. We also see how in particular in the case of backward motion, the correct body pose and gaze angles have been maintained. Fig.97(c) also shows how the system can correctly track sideway glances where the motion and gaze angles differ significantly during tracking.

We next demonstrate the system with multiple individuals. Fig.98 shows several examples of two people standing closely, where the system correctly estimates the body pose and gaze angles for different situations. Fig.99 shows an experiment with three people moving and turning freely.

We performed a quantatively evaluation based on ground truth on all three sequences in Table 10. Pose and gaze groundtruth was annotated in 3D using a customized user interface. We compare our method against two methods: *(1)* a baseline method which uses a simple groundplane person traker similar to Fig.96, and *(2)* a gaze tracking method reported in [86] which uses a separate person and gaze tracker. For the baseline method, gaze and pose was assumed to be equivalent to motion direction. The method in [86] provided gaze information but not the body pose, so we again equate the pose to motion direction. In five out of six cases, our method has the smallest estimation error. The advantage of correctly modeling the relationship between motion direction,

(a)

(b)

(c)

Figure 97: Pose and gaze estimation with face detections from PTZ cameras: (a) forward and (b) backward motion, (c) sideways glance with large difference between body pose and gaze. Overall tracking improves significantly when compared to Fig.96.

body pose, and gaze is clear.

Table 10: Average angle difference in degrees for our method, the gaze tracker, and the baseline tracker.

	Our method		Gaze tracker [86]		Baseline	
	pose	gaze	pose	gaze	pose	gaze
one person	**19.47**	**23.82**	57.58	33.01	39.84	33.10
two people	**57.12**	**35.65**	73.40	45.33	73.40	88.32
three people	**42.40**	38.30	73.55	**37.31**	73.55	90.67

The above experiments demonstrate the efficacy of estimating gaze and body pose for a group of people in close proximity. We foresee that the system should extend well to work in a crowded condition, provided that a sufficient number of PTZ views are added.

Figure 98: Pose and gaze estimation of two people (a) facing each other, (b) standing next to each other, and (c) facing in opposite directions. The top second view in each case depicts an artificial top-down re-rendering, which provides a better visualization of the relative pose and gaze.

F.6 Discussion and Conclusions

We have presented a comprehensive system for tracking location, body pose and gaze direction in unconstrained environment using surveillance and PTZ cameras. We have shown through qualitative experiments that the system performs well under a variety of experimental conditions. The algorithm has important applications in security, surveillance, and behavior recognition, as well as the emerging areas of gaming, interactive entertainment and advertising.

Acknowledgement. This project was supported by grants #2007-RG-CX-K015 and #2009-SQ-B9-K013 awarded by the National Institute of Justice, Office of Justice Programs, US Depart-

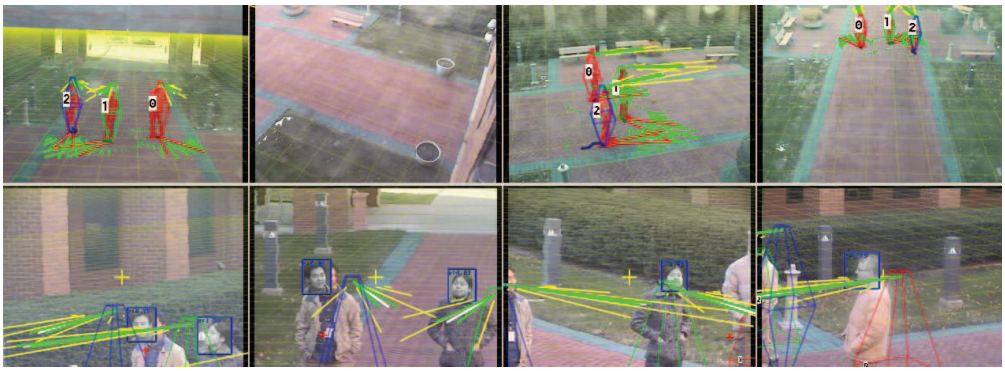


Figure 99: Pose and gaze estimation for three people moving and turning freely.

ment of Justice. The opinions, findings, and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of the Department of Justice.

G Group Context Learning for Event Recognition

The following is a reprint of a manuscript that is submitted to the IEEE Workshop on Applications of Computer Vision (WACV) 2012. The precise title of the paper is "Group Context Learning for Event Recognition". This work is done by our intern researcher Yimeng Zhang during the summer of 2011. Yimeng is a Ph.D. student from the Department of Electrical and Computer Engineering, Cornell University. She has been working on machine learning based event classification such as using spatial temporal histograms and bag of visual words.

Abstract

We address the problem of group-level event recognition from videos. The events of interest are defined based on the motion and interaction of members in a group over time. Example events include group formation, dispersion, following, chasing, flanking, and fighting. To recognize these complex group events, we propose a novel approach that learns the group-level scenario context from automatically extracted individual trajectories. We first perform a group structure analysis to produce a weighted graph that represents the probabilistic group membership of the individuals. We then extract features from this graph to capture the motion and action contexts among the groups. The features are represented using the "bag-of-words" scheme. Finally, our method uses the learned Support Vector Machine (SVM) to classify a video segment into the six event categories. Our implementation builds upon a mature multi-camera multi-target tracking system that recognizes the group-level events involving up to 20 individuals in real-time.

G.1 Introduction

Recognizing events of interest from surveillance videos is an important topic and has been extensively studied. Applications include monitoring transportation hubs, public venues, and yards for security and safety. In general, the efforts can be organized into three main categories: *(i) action recognition* [96, 97, 98, 99], such as recognizing if a person is walking or chatting, where the analysis of the body articulation is essential; *(ii) interaction recognition* [100, 101, 102] between

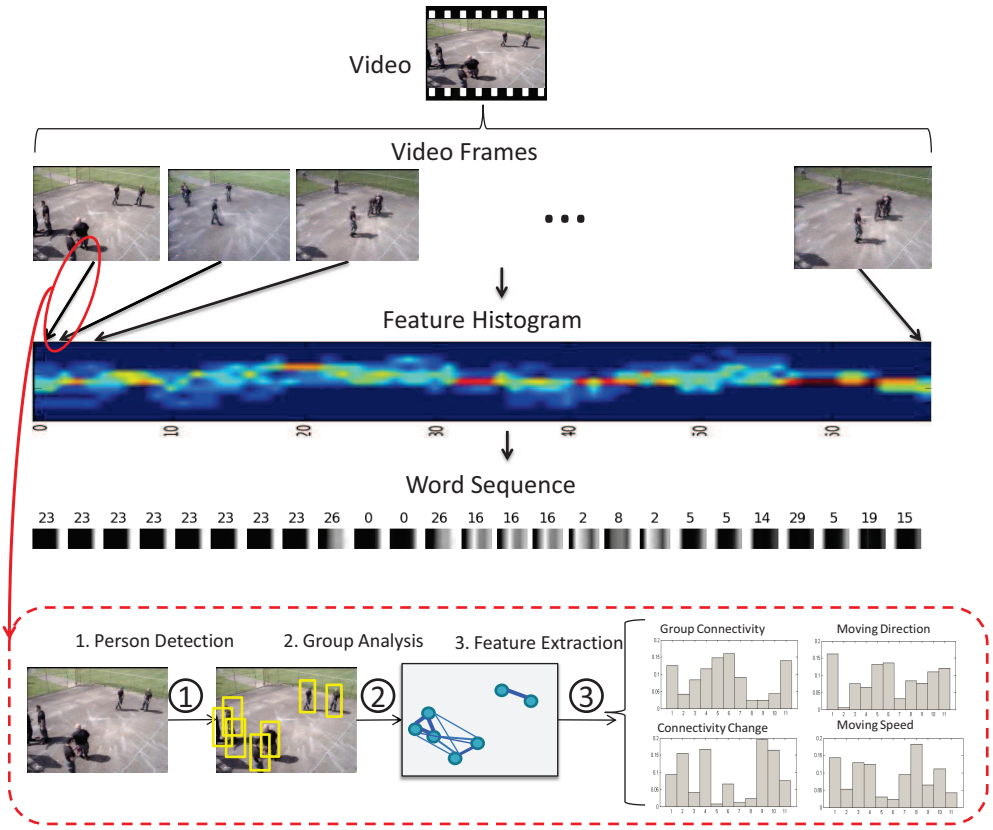


Figure 100: **Overview: Learning group context words.** Given a video we extract feature histograms for each frame to represent their group connectivity and motion features. To extract features from a frame, we go through three steps as illustrated at the bottom of the figure. The feature histogram image shown is column-wise stacked, where the red indicates large value and blue indicates small value. Group context words learned from the feature histograms are visualized in the middle, where each word is created with the histogram of four consecutive frames, and each row in a word depicts a histogram for a frame.

a few individuals or with respect to an object, such as determining if two people are meeting or exchanging items, where both the overall motion and articulation are useful cues; and *(iii) crowd event recognition* [103, 104, 105, 106, 107, 108, 109, 110], such as detecting abnormal traffic or aggressive fight involving groups of individuals, where the scenario is most complicated, since a diverse range of activities could occur in a crowded scene.

In this paper, we are interested in recognizing events involving groups of people and the interaction among them. Example scenarios including group dynamic analysis such as group formation, dispersion, and one or more groups approaching another group. We are also interested in detecting group-level behaviors such as chasing and aggression among groups, which are likely related to

(a) group dispersion (b) group formation (c) approaching (d) group flanking (e) group fighting

Figure 101: Example scenarios we aim to recognize from videos. Typical behaviors of interest include group formation and dispersion, a group approaching, following or chasing another group, one or more groups aggressively surrounding another group (flanking), which likely suggests a fight, and the actual group fighting.

potential fighting and security concerns. Our task is different from those crowd event recognition tasks that are mainly based on motion analysis of dense crowds [111, 105, 107, 109], and focus on the macro-analysis of the crowd rather than the micro-dynamics of groups and the interaction of individuals. Figure 101 illustrates typical scenarios that we aim to recognize from surveillance videos.

We propose a novel learning-based framework for group event recognition that automatically learns the *group context* for different event categories. The group context refers to the group-level interactions among people over time. An overview of our approach is illustrated in Figure 100. We first detect and track people in the video using standard methods [112, 113]. Based on the tracking, we analyze the group-level structure and motion, and then extract the group context features based on the probabilistic (soft) group structure analysis results (Figure 102). We can then recognize group-level behaviors and detect events of interest using the learned group context features.

A key step towards robust event detection is the ability to recognize the *temporal co-occurrence* of similar group context patterns that appear in videos, which can occur at different locations, scales, and times. To illustrate the challenge, a video containing group formation could possibly involve a variable number of people getting together in a variable length of time. Thus, the occurrence and co-occurrence of different group context patterns are the key to recognize this particular scenario. In order to achieve robust event recognition, we first define several robust features that model the interaction and motion between individuals among the groups (group context), which can be detected on a per-frame or per-segment basis (Figure 100). Second, to capture the temporal co-occurrences of these features exacted from different frames in a video, we adopt the "bag-of-

words" scheme [114] by clustering them into group context words. Finally we train a SVM with bags of group context words to classify the videos into different event categories.

We summarize our contributions as follows: (1) we develop a novel machine learning based framework to robustly recognize group-level events from videos; (2) we propose robust features that model the group context of individuals with motion tracking; and (3) we implement our algorithm with a multi-camera tracking system and demonstrate it in a real-time event recognition system in surveillance applications.

G.2 Related Works

With the prevalence of surveillance cameras, event recognition has drawn increasing attention from the computer vision community. Some works consider the problem of recognizing actions performed by a single person, such as [96, 97, 98, 99, 115, 116, 117]. Activity categories such as walking or running are defined and detected straight from analyzing body part movements of a person.

More relevant to our work is the recognition of events involving multiple agents in a crowd scene. Existing works typically focus on events defined by the movements of individuals or the entire crowd. Typical applications include the detection of cars or people with abnormal movements in a traffic scene. There are mainly two types of approaches in this category. The first type is *object centric* [111, 118], where the trajectories of detected targets are analyzed for recognition. The second type is *view or flow centric* [103, 104, 105, 106, 107, 108, 109, 110], which avoids object tracking, and instead models the crowd motions with dense optical flows, or the gradients and appearances of the spatio-temporal subvolumes. In particular, Zen and Ricci [104] first cluster low level optical flow features into atomic events, and then learn the salient activity patterns from histograms of atomic activities through a convex optimization problem. Tran and Yuan [103] follow a spatio-temporal subvolume search scheme for abnormal event detection and localization. A message passing algorithm is developed to find the global optimal path *w.r.t.* the 3D locations of the subvolumes in the video. Saxena *et al.* [119] trigger abnormal crowd events by imposing a hard

threshold on several measures including crowd density, principal directions, number of individual motion vectors in a crowd. The non-object centric approach is also popular for *general event or action categorization* from movies or youtube videos [120, 121, 122]. Motion or appearance features are extracted from spatio-temporal subvolumes detected at interest points. Usually a visual word representation is followed for the final event categorization.

We focus on the type of events that are defined not only by the motion information but also by the interactions of groups or the individuals among groups, such as "group fighting" and "flanking" (a group of people surrounding another group). Previous works on this type of events mainly use the logic or rule based methods, which require manual creation of rules [112, 123, 124, 101, 102]. Chang *et al.* [112] recognize various group events by combining the results from probabilistic group structure analysis and motion analysis and checking against a list of event models, which are defined manually using scenario-specia predicates. To explicitly model the temporal constraints pertain to complex events, different probabilistic logical inference engines have been built, such as the Markov Logic Networks [102] and probabilistic event logic [123]. These works use rule or logic based approach, and thus require experts to manually create person-person or person-object rules with domain knowledge. Therefore, the performance of event recognition highly depends on how well the rules are defined. The learning curve for a general operator to define a set of compatible rules could be sharp. Moreover, the rules in these methods often rely on clean input observations, which are hardly the case for results obtained from automatic detection and tracking algorithms.

The learning-based approach of Choi *et al.* [125, 126] is relevant to us. They focus more on atomic actions such as people queuing and talking, thus only human pose and spatial distance cues are considered for event recognition. In comparison, our events of interest involve group context and require further analysis on the group-level motion and interaction cues that could possibly change over time.

G.3 Approach

We propose to extract robust group context features from video and adopt a bag-of-words learning scheme to recognize group-level events. Figure 100 illustrates our overall approach. Given an input video segment, we first perform person detection and tracking. We then perform group structure analysis of the tracked individuals, as a mean to extract group context features. Following [112], we retain a probabilistic group representation, such that the group-level information can be reliably captured. Specifically, the group analysis produces a weighted connectivity graph for each frame, where the nodes of the graph are the detected individuals and the weight of an edges is the probability of two individuals being in the same group. From the connectivity graph, we extract features that capture pair-wise group relationships among the individuals and their motion information. Finally we cluster the extracted features into group context words and use "bag of group context words" to train a SVM to classify the input video segment into an event category. In the following sections, we will explain the details of each step in our approach.

G.3.1 Video Tracking System

We briefly explain the multi-view, multi-target tracking system that is used as a baseline component. Note that the event recognition algorithms introduced in this paper is general and can be applied to tracking results from other systems.

We take the videos from three standard CCTV cameras of overlapping views. All cameras are calibrated and synchronized. Figure 102 gives a snapshot of our system in operation, where the movements of the individuals are tracked cooperatively across cameras. Person detections from each view are projected onto the ground plane in 3D and fed into a centralized tracker which is implemented with a Kalman filter.

G.3.2 Group Analysis

Given the tracking of individuals, we incorporate a probabilistic grouping strategy similar to [112] to perform group analysis and to extract group-level motion and interaction features. As opposed

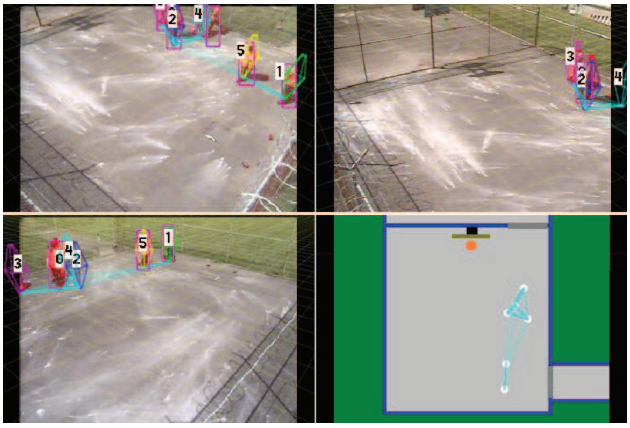


Figure 102: Video tracking system performing person detection and tracking from one or more views. A top-down synthesized view is generated to visualize the probabilistic grouping connectivity w_{ij} between individuals under tracking.

to approaches that rely on hard, agglomerative or decisive clustering techniques to define groups, the probabilistic grouping without a hard segmentation of groups keeps more reliable information about the dynamics of groups that will be later on used to calculate the group context features.

For each frame t, we define a connectivity graph $\mathcal{G}^t$ to represent the connectivity (or the probability) of two individuals i and j belonging to a group at frame time t. Specifically, for each edge e_{ij}^t in $\mathcal{G}^t$, the edge weight w_{ij}^t represents the probability that individuals i and j belong to the same group, $0 \leq w_{ij}^t \leq 1$. The definition of the connectivity w_{ij}^t is motivated by two lines of thoughts: (1) a *track-to-track* connectivity that considers the motion of the two individuals i and j under tracking, including the spatial distance and moving direction calculated from a small period of time in the tracking history, and (2) a *path-based* connectivity that considers the existence of neighboring individuals that increase the overall grouping strength of nearby individuals all together. The bottom-right image in Figure 102 illustrates an example of the probabilistic grouping graph $\mathcal{G}^t$ from a synthesized top-down view of the tracked individuals.

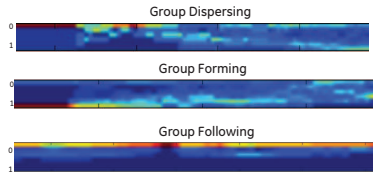
G.3.3 Feature Extraction

In order to achieve the recognition of group-level events that could occur in variable time scales, we propose to use robust features that can be efficiently extracted and can capture information about the group structure, motion, and dynamics. Our solution is to extract the following four

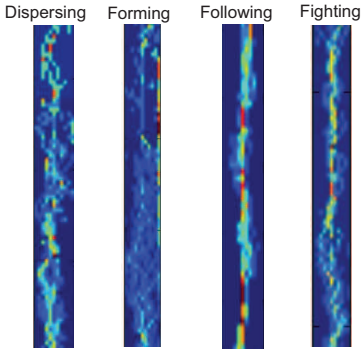
types of features: (1) group connectivity, (2) connectivity change, (3) motion direction, and (4) motion speed (Figure 103).

Group connectivity. This feature models the group structures in a frame t by creating a histogram of the edge weights w_{ij}^t from the group connectivity graph $\mathcal{G}^t$. The histogram is normalized by the total number of edges so that the number of people will not bias the measurement. Figure 103(a) shows the group connectivity histograms of example videos from four event categories (group dispersion, formation, following, and fighting), where each row depicts a histogram at a frame, and the column axis indicates time. Observe that in the beginning of a group dispersing event, the bins correspond to high connectivity values have more counts, and as the event unfolds in time, the bins of low connectivity values receive more counts. In other words, the strength of group connectivities decreases over time. In the contrary, for the group forming event, the strength of group connectivity increases over time. For the group following and fighting events, people mostly have high connectivity values, since people maintain tight groups in these events. These observations verify that this novel feature captures discriminative cues to distinguish various group-level events.

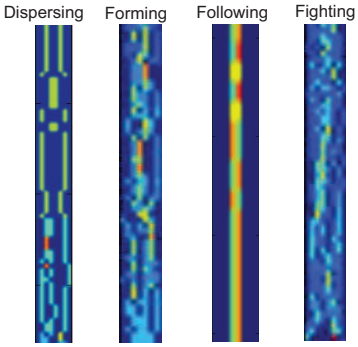
Connectivity change. This feature models the group connectivity change between the current frame t and a previous frame t' ($t' = t - 1/S$ second), for each pair of individuals i and j who are detected in both frames. The connectivity difference for i and j is $\Delta_{ij}^t = w_{ij}^t - w_{ij}^{t'}$, where w_{ij}^t is the weight for the edge between individual i and j in $\mathcal{G}^t$. The connectivity change feature is the histogram of such differences of all person pairs in the current frame, and again the histogram is normalized by the total number of edges. Figure 103(b) shows the connectivity change histograms for four event categories. The center of a histogram represents no connectivity change ($\Delta_{ij}^t \approx 0$); bins to the left correspond to negative changes ($w_{ij}^t < w_{ij}^{t'}$), that is, the group connectivity of the current frame is smaller than the one in the previous frame; bins to the right correspond to positive changes. The figure shows that the connectivity changes are mostly negative for the group dispersing event and positive for the group forming event. The connectivity change is almost 0 for the group fighting and following events since the group structures are reasonably stable during



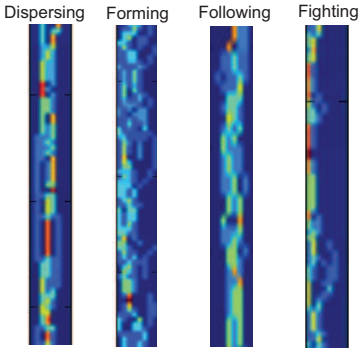
(a) Group connectivity



(b) Connectivity change



(c) Moving direction



(d) Moving speed

Figure 103: Feature histograms for example videos from four event categories. Red denotes higher value, and blue denotes lower value. See text for explanation of how the histograms capture group context features.

these events.

Motion direction. We also record the moving direction of each person i by the velocity direction $d_i^t \in [0, 2\pi)$. To deal with camera rotations and view point changes, we normalize the direction d_i^t of each person by subtracting the mean of them d_μ^t in the $[0, 2\pi)$ periodic space: $\hat{d}_i^t = d_i^t - d_\mu^t$. This normalized motion direction $\hat{d}_i^t$ is used to compute the motion direction histogram. Figure 103(c) shows example motion direction histograms for different event categories. In a group following event, people tend to have similar motion directions, since that direction is the one all are heading towards. For other events, the moving directions have a wider distribution.

Motion speed. This feature captures the motion speed s_i^t (magnitude of velocity) for each person i. Observe in Figure 103(d) that people do not move much when they are engaged in a fight; while for the dispersion and following events, people show larger motion speeds over time.

All four features can be extract directly and efficiently from trajectories obtained from the tracker. This features robustly capture group structure and dynamic changes over time. We will next describe how we formulate our bag-of-words learning scheme based on these features.

G.3.4 Learning Group Context Words

After feature extraction, a video can be represented as a sequence of feature histograms (Figure 100). Direct learning of classifiers on these sequences is difficult, especially when we have a long video. Moreover, the video of an event can have variable lengths, and the starting and ending time of the event are unknown, rendering the problem more difficult. We propose to cluster the feature histograms into a few representative clusters, which we refer to as *group context words*.

To create such words, we first represent each frame by concatenating the histograms of the current and previous consecutive T frames. This concatenated histogram models local histogram changes and smoothes out the noise in the observations from a single frame. Then we cluster the concatenated histograms using K-means into a vocabulary of $|V|$ words. A word represents a certain pattern of the local histograms. Since we have four types of features, we create a vocabulary for each feature type. Thus for each feature type, a video will be represented as a sequence of

words.

We adopt the "bag-of-words" model, which represents a video as a histogram of words. We create a bag-of-words histogram for each feature type and concatenate them together. These concatenated word histograms are used to train a SVM to classify the input video into different event categories.

G.4 Experiments

We evaluate our approach on part of the Mock Prison Riot (MPR) dataset (http://mockprisonriot.org) as in [112]. The dataset has 19 surveillance videos taken in an abandoned prison yard in West Virginia. In these videos, several volunteer correction officers enact typical behaviors of interest of the prisoners. The length of each video varies from 3 to 6 minutes. Example snapshots of the dataset can be found in Figure 101. We report our performance for the following six categories of group-level events: (1) group formation, (2) group dispersion, (3) group following, (4) group chasing, (5) group flanking, and (6) group fighting.

G.4.1 Event Recognition

The first experiment we performed is to classify an entire input video into one of the six predefined event categories, *i.e.*, to determine whether an event occurs or not. For this experiment, we manually segmented the videos in the dataset into 177 non-overlapping small video segments of 2 to 30 seconds. For each segment, we label all events occurred. Some events may overlap with others and occur at the same time. If no events of interest occurred in a segment, we label it as "random", which serves as negative examples for all other categories. Note that we do not need to label a clear start and end point for each event in the video. We randomly select 60% of the videos for training and the rest for testing.

We illustrate the words of the "connectivity change" feature that occur most frequently in the videos of the four event categories (dispersing, forming, following, and fighting) in Figure 104. As described in Section G.3.4, a word is constructed with the histograms of consecutive T frames (we

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

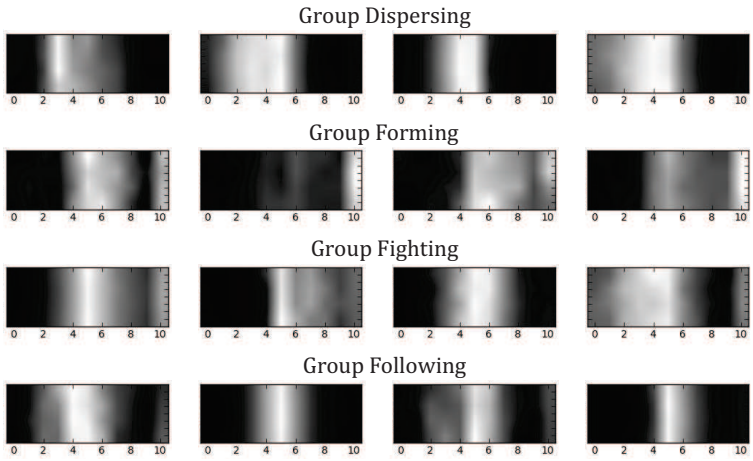


Figure 104: The "connectivity change" words that occur most frequently in the videos of different event categories. Each word is created with the histograms from four consecutive frames. For each word image, one row correspondes to a histogram exacted from one frame.

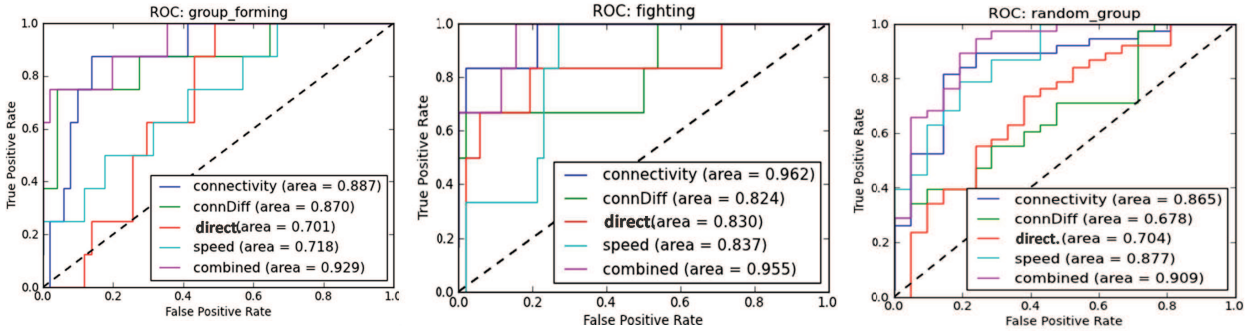


Figure 105: The ROC curves using different types of features for event categories: (left) group forming, (middle) group fighting, and (right) random group. The "random group" ROC curve shows the performance for classifying the events of interest vs. normal behaviors.

use $T = 4$). In the figure, each row of a word image represents a "connectivity change" feature histogram exacted from a frame. Notice these feature histograms show similar observations as those visualized in Figure 103(b). The most frequent words for the group dispersing event are those that represent the frames where the group connectivities among people decrease as compared to the previous frames. On the contrary, the top words for the group forming event are those that show the opposite pattern. For the group fighting and group following events, the top words correspond to the histograms where the group connectivity change is close to zero.

We train an one-vs-all SVM for each event category, and evaluate the recognition performance with the ROC curves drawn with the probabilistic scores generated by the SVMs. Figure 105 shows

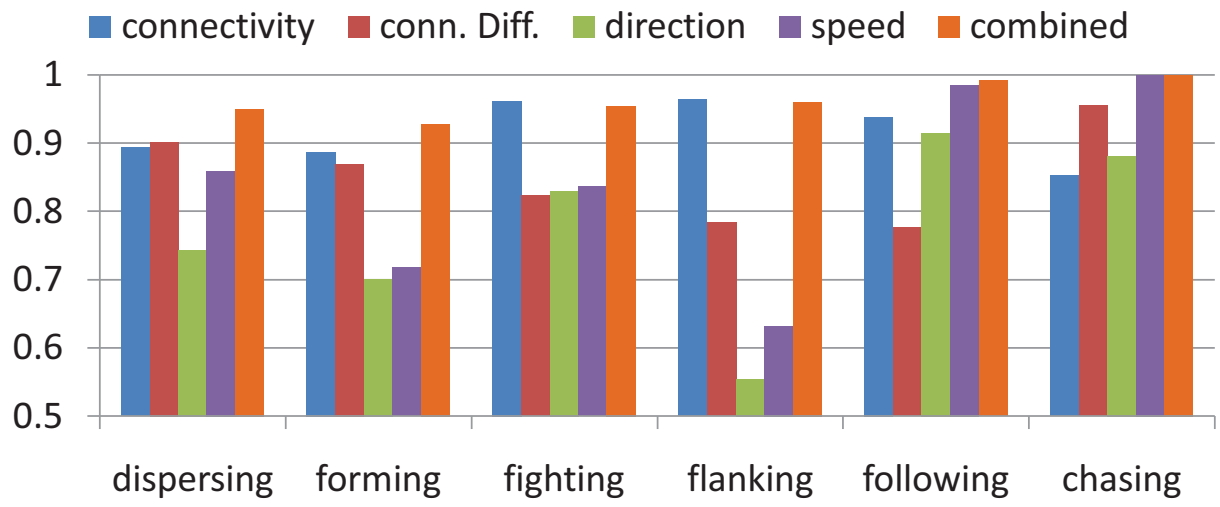


Figure 106: The AUC scores of the ROC curves for different event categories using different types of features.

the ROC curve for three example event categories using different feature types. The "combined" one uses all four feature types concatenated as described in Section G.3.4. As shown in the figure, for the group forming event, the group connectivity and connectivity change features are more useful, compared to the motion direction and speed features. While for the group fighting event, the group connectivity feature outperforms other feature types. The combined one achieves the best performance. We also show the ROC curve for the random group, which indicates the performance for distinguishing behaviors of interests from normal behaviors. We show the AUC (area under curve) scores for all categories in Figure 106. Similar to the results for the forming event, "group connectivity" and "connectivity change" features are more discriminative for recognizing group dispersing. The speed feature performs better for group chasing events, since people move fast in the chasing events, which is a very distinct feature for this particular event category as compared to the other five. The speed feature also works well for the following category, since people keep relatively constant speed over time when following each other. The performance with the combined features is the best for all event categories except for the fighting category, where the combined feature performs worse but still comparable to the connectivity feature. We achieve more than 90% AUC scores for all categories.

G.4.2 Event Detection

In the second experiment we perform online event detection, that is, to determine whether any event of interests occurs at each frame in the input video. We label the start and end points of the occurred events for the videos in our dataset. This scenario is useful to provide real-time alerts to the operators. Since clear start and end points are difficult to determine for several events, we label the one second period around the start and end points of an event as ambiguous frames, and do not use them for evaluation. We randomly select 60% of the 19 videos in the dataset for training, and the rest for testing. We make prediction at every 4^{th} frame in a video, using observations from a four-second temporal window ($[t - 4s, t]$), *i.e.*, the previous 4 seconds. Other aspects of the algorithm remain the same.

We compare the performance of our approach against the state-of-the-art approach introduced in [112], which adopts a rule-based method for event detection. Probabilistic rules are created manually for each event category and probabilistic decisions are made at each frame. Figure 107 shows the prediction results on an example test video. The rule-based method generates much more false positives than our method. One main reason is that the rule-based method is more sensitive to the person detection errors, since the errors are not considered when creating the rules, whereas our method can tolerate more observation noise by learning from the training data. The other reason is that the rules in [112] only consider the past several frames when making the decision for the current frame, while our method uses a much larger temporal window (4 seconds) and is thus able to remove some local noise. Table 11 shows the AUC score comparison for different event categories. Our method outperforms the rule-based method [112] on all categories. Note that the "fighting" event is not defined in [112], since its occurrence in a video usually spans a long duration. As we already discussed, the rules in [112] are usually defined with only a few frames.

We implement our approach with C++ and Python. On an Intel 2.4G dual-core computer, the entire event detection system, including person detection and tracking, takes around 0.02 second to process a 640x480 frame. Therefore, we can detect events of interest in real-time.

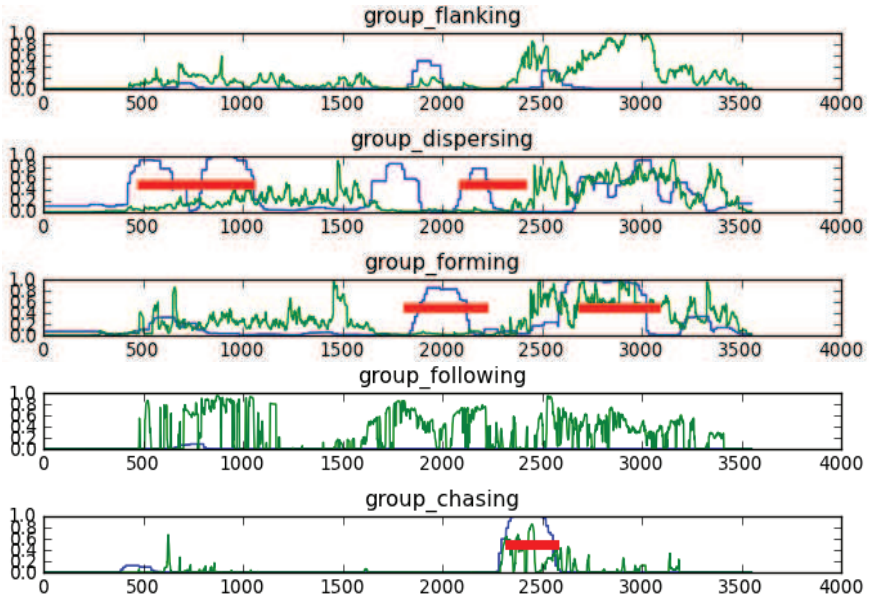


Figure 107: The predicted probabilities for a test video. Red lines represent the ground truth labels. Green lines denote the probabilities at each frame using the rule-based method [112]. Blue lines denote the probabilities using our approach.

	Dispersing	Forming	Flanking	Following	Chasing	Fighting
rule-based	0.592	0.658	0.921	0.667	0.981	N/A
ours	**0.950**	**0.929**	**0.961**	**0.992**	**1.000**	**0.955**

Table 11: The AUC scores of the ROC curves using the rule-based method [112] and our method.

G.5 Conclusion

We proposed a novel learning based framework for group-level event recognition. Unlike most existing event recognition works, which define the events based on the movements of an individual or the entire crowd, the events discussed in this paper focus more on the interactions among people. We designed robust features that can capture the group context of individuals in a video. We built a system with the proposed algorithm, which can process a video and detect the events in real-time. The performance of the system significantly outperforms the state-of-the-art method on a challenging dataset.

Future work. First, we would like to explore more interesting event categories. Second, we are interested in developing algorithms that can combine different types of features in a more sophisticated way, rather then concatenating them with the same weights. Finally, instead of the "bag-of-words" method, we plan to develop algorithms that can model the temporal order among the words.

References

[1] G. Klyne and J. J. Carroll. Resource description framework (RDF): Concepts and abstract syntax, February 2004.

[2] Edward T. Hall. *The Hidden Dimension*. Anchor, 1966.

[3] D. Helbing and P. Molnar. Social force model for pedestrian dynamics. *Physical Review E*, 51:4282–4286, 1995.

[4] Ming-Ching Chang, Nils Krahnstoever, and Weina Ge. Probabilistic group-level motion analysis and scenario recognition. In *To appear in ICCV*, 2011.

[5] M. Zweig and G. Campbell. Receiver-operating characteristic (ROC) plots: A fundamental evaluation tool in clinical medicine. *Clinical Chemistry*, 39:561–Ű577, 1993.

[6] T. Fawcett. An introduction to ROC analysis. *Pattern Recognition Letters*, 27:861–Ű874, 2006.

[7] M. Jordan. *Graphical Probability Models*. In publication.

[8] F. V. Jensen. *Bayesian Networks and Decision Graphs*. Springer, 2001.

[9] Li Ding. *Enhancing Semantic Web Data Access*. PhD thesis, University of Maryland, Baltimore County, April 2006.

[10] Frank Manola and Editors Eric Miller. Rdf primer, w3c recommendation. http://www.w3.org/TR/2004/REC-rdf-primer-20040210/ . Latest version available at http://www.w3.org/TR/rdf-primer/, February 2004.

[11] Alan Yuille and Hongjing Lu. The noisy-logical distribution and its application to causal inference. 2009.

[12] Stuart J. Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach (2nd Edition)*. Prentice Hall, 2002.

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

[14] J. P. Lewis. Fast normalized cross-correlation. *Vision Interface*, pages 120 – 123, 1995.

[15] Robert Sedgewick. *Algorithms in C++*. Addison-Wesley Professional, second edition, 1992.

[16] Jorge M. Santos, Joaquim Marques de Sa, and Luis A. Alexandre. Legclust—a clustering algorithm based on layered entropic subgraphs. *PAMI*, 30(1):62–75, 2008.

[17] Hong Chang and Dit-Yan Yeung. Robust path-based spectral clustering. *Pattern Recogn.*, 41(1):191–203, 2008.

[18] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. John Wiley and Sons, Inc., second edition, 2001.

[19] Ming-Ching Chang, Nils Krahnstoever, Sernam Lim, and Ting Yu. Group level activity recognition in crowded environments across multiple cameras. In *AVSS W. AMMCSS*, 2010.

[20] Ting Yu, Sernam Lim, Kedar Patwardhan, and Nils Krahnstoever. Monitoring, recognizing and discovering social networks. In *CVPR*, 2009.

[21] D. Koller and N. Friedman. *Bayesian Networks and Beyond*. In publication.

[22] Vibhav Gogate and Rina Dechter. Approximate inference algorithms for hybrid Bayesian networks with discrete constraints. In *Proc. Uncertainty in Artificial Intelligence*, pages 209–216, Arlington, Virginia, 2005. AUAI Press.

[23] Helge Langseth, Thomas D. Nielsen, Rafael Rumi, and Antonio Salmeron. Inference in hybrid Bayesian networks. *Reliability of Engineering and System Safety*, 94:1499–1509, 2009.

[24] M. Neil, M. Tailor, M. D. Marquez, N. Fenton, and P. Hearty. Modelling dependable systems using hybrid bayesian networks. *Reliability Engineering & System Safety*, 93(7):933–939, 2008.

[25] Mark Lutz. *Programming Python*. O'Reilly Media, 2006.

[26] (PI) N. Krahnstoever. Automatic detection and prevention of disorderly and criminal activities. Grant #2007-RG-CX-K015, 2008. National Institute of Justice.

[27] Raed Shatnawi, Wei Li, James Swain, and Tim Newman. Finding software metrics threshold values using ROC curves. *Journal of Software Maintance and Evalution: Research and Practice*, 22:1–16, 2010.

[28] Yimeng Zhang and Tsuhan Chen. Efficient kernels for identifying unbounded-order spatial features. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1762 –1769, 2009.

[29] J. F. Allen and G. Ferguson. Actions and events in interval temporal logic. *Journal of Logic and Computation*, 27:861–874, 1994.

[30] Boris Lau, Kai O. Arras, and Wolfram Burgard. Multi-model hypothesis group tracking and group size estimation. *International Journal of Social Robotics*, 2009.

[31] Weina Ge, Robert T. Collins, and Barry Ruback. Automatically detecting the small group structure of a crowd. In *WACV*, pages 1–8, 2009.

[32] Shobhit Saxena, François Brémond, Monnique Thonnat, and Ruihua Ma. Crowd behavior recognition for video surveillance. In *ACIVS*, pages 970–981, Berlin, Heidelberg, 2008. Springer-Verlag.

[33] M. Girvan and M. E. Newman. Community structure in social and biological networks. *Proc Natl Acad Sci U S A*, 99(12):7821–7826, June 2002.

[34] M. E. J. Newman. Finding community structure in networks using the eigenvectors of matrices. *Physical Review E (Statistical, Nonlinear, and Soft Matter Physics)*, 74(3):036104, 2006.

[35] Andreas Ess, Bastian Leibe, Konrad Schindler, and Luc van Gool. Robust multiperson tracking from a mobile platform. *IEEE Trans. Pattern Anal. Mach. Intell.*, 31(10):1831–1846, 2009.

[36] 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.

[37] Biswajit Bose, Xiaogang Wang, and Eric Grimson. Multi-class object tracking algorithm that handles fragmentation and grouping. In *IEEE CVPR*, 2007.

[38] 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 *Proc. Ninth IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETS), New York*, 2006.

[39] P. Tu, T. Sebastian, G. Doretto, N. Krahnstoever, J. Rittscher, and T. Yu. Unified crowd segmentation. In *Proceedings of European Conference on Computer Vision*, 2008.

[40] Ting Yu, Sernam Lim, Kedar Patwardhan, and Nils Krahnstoever. Monitoring, recognizing and discovering social networks. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[41] 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.

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

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

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

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

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

[47] 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.

[48] Behave dataset. *http://homepages.inf.ed.ac.uk/rbf/BEHAVE/*.

[49] Preben Fihl and Thomas B. Moeslund. Pose estimation of interacting people using pictorial structures. In *In Proc. Seventh IEEE International Conference on Advanced Video and Signal Based Surveillance*, 2010.

[50] Xiaoming Liu, N. Krahnstoever, Ting Yu, and Peter Tu. What are customers looking at? In *Proc. IEEE International Conference On Advanced Video and Signal Based Surveillance*, 2007.

[51] Ming-Ching Chang, Nils Krahnstoever, Sernam Lim, and Ting Yu. Group level activity recognition in crowded environments across multiple cameras. In *In Proc. Seventh IEEE International Conference on Advanced Video and Signal Based Surveillance, Workshop on Activity Monitoring by Multi-Camera Surveillance Systems (AMMCSS)*, 2010.

[52] Raymond Ptucha and Andreas Savakis. Facial pose estimation using a symmetrical feature model. In *Proc.Int'l Conference on Multimedia and Expo.*, pages 1664–1667, 2009.

[53] Erik Murphy-Chutorian and Mohan Trivedi. Head pose estimation in computer vision: a survey. *PAMI*, 31(4):607–626, 2009.

[54] Proceedings of CLEAR'07 workshop: Classification of events, activities and relationships (http://www.clear-evaluation.org). In *LNCS*, 2007.

[55] CLEAR: Classification of Events, Activities and Relationships (http://www.clear-evaluation.org). 2006, 2007.

[56] Kohsia S. Huang and Mohan M. Trivedi. Robust real-time detection, tracking, and pose estimation of faces in video streams. In *ICPR*, pages 965–968, 2004.

[57] Martin Bäuml, Keni Bernardin, Mika Fischer, and Hazim Kemal Ekenel. Multi-pose face recognition for person retrieval in camera networks. In *AVSS*, 2010.

[58] Oswald Lanz and Roberto Brunelli. Joint bayesian tracking of head location and pose from low-resolution video. In *LNCS (CLEAR'06)*, volume 4625, pages 287–296, 2008.

[59] Changbo Hu, Jing Xiao, Iain Matthews, Simon Baker, Jeffrey Cohn, and Takeo Kanade. Fitting a single active appearance model simultaneously to multiple images. In *BMVC*, 2004.

[60] Michael Voit, Kai Nickel, and Rainer Stiefelhagen. Head pose estimation in single- and multi-view environments – results on the CLEAR'07 benchmarks. In *CLEAR*, 2007.

[61] C. Canton-Ferrer, J.R. Casas, and M. Pardàs. Head orientation estimation using particle filtering in multiview scenarios. In *CLEAR*, pages 305–310, 2007.

[62] N. Krahnstoever, Ting Yu, Ser-Nam Lim, Kedar Patwardhan, and Peter Tu. Collaborative real-time control of active cameras in large scale surveillance systems. In *Proc. Workshop on Multi-camera and Multi-modal Sensor Fusion Algorithms and Applications (M2SFA2)*, October 2008.

[63] J. Munkres. Algorithms for the assignment and transportation problems. *Journal of SIAM*, 5:32–38, 1957.

[64] H. Schneiderman. Learning a restricted Bayesian network for object detection. In *CVPR*, pages 639–646, 2004.

[65] Pittpatt: Pittsburgh pattern recognition. *www.pittpatt.com*.

[66] Xiaoming Liu. Discriminative face alignment. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 31(11):1941–1954, November 2009.

[67] Xiaoming Liu. Video-based face model fitting using adaptive active appearance model. *Image and Vision Computing*, 28(7):1162–1172, July 2010.

[68] Joshua Candamo, Matthew Shreve, Dmitry B. Goldgof, Deborah B. Sapper, and Rangachar Kasturi. Understanding transit scenes: a survey on human behavior-recognition algorithms. *ITSS*, 11(1):206–224, 2010.

[69] Beibei Zhan, Dorothy N. Monekosso, Paolo Remagnino, Sergio A. Velastin, and Li-Qun Xu. Crowd analysis: a survey. *Mach. Vis. Appl.*, 19(5/6):345–357, 2008.

[70] Bingbing Ni, Shuicheng Yan, and Ashraf Kassim. Recognizing human group activities with localized causalities. In *CVPR*, pages 1063–6919, 2009.

[71] Anthony Hoogs, Steve Bush, Glen Brooksby, Amitha Perera, Mark Dausch, and Nils Krahnstoever. Detecting semantic group activities using relational clustering. In *WMVC*, pages 1–8, 2008.

[72] Carolina Garate, Piotr Bilinski, and Francois Bremond. Crowd event recognition using HOG tracker. *PETS*, pages 1–6, 2009.

[73] M. Ryoo and J. Aggarwal. Stochastic representation and recognition of high-level group activities. *IJCV*, pages 1–18, 2010.

[74] Shaogang Gong and Tao Xiang. Recognition of group activities using dynamic probabilistic networks. In *ICCV*, pages 742–749, 2003.

[75] S. Pellegrini, A. Ess, K. Schindler, and L. van Gool. You'll never walk alone: Modeling social behavior for multi-target tracking. In *ICCV*, pages 261–268, 2009.

[76] Ramin Mehran, Alexis Oyama, and Mubarak Shah. Abnormal crowd behavior detection using social force model. In *CVPR*, pages p35–942, 2009.

[77] S. Ali and M. Shah. Floor fields for tracking in high density crowd scenes. In *ECCV*, pages II: 1–14, 2008.

[78] Mikel Rodriguez, Saad Ali, and Takeo Kanade. Tracking in unstructured crowded scenes. In *ICCV*, pages 1389–1396, 2009.

[79] Ran Eshel and Yael Moses. Tracking in a dense crowd using multiple cameras. *IJCV*, 88(1):129–143, 2010.

[80] F. Cupillard, F. Bremond, and M. Thonnat. Group behavior recognition with multiple cameras. In *WACV*, pages 177–183, 2002.

[81] V. Mahadevan, W. Li, V. Bhalodia, and N. Vasconcelos. Anomaly detection in crowded scenes. In *CVPR*, 2010.

[82] Kevin Smith, Sileye O. Ba, Jean-Marc Odobez, and Daniel Gatica-Perez. Tracking the visual focus of attention for a varying number of wandering people. *PAMI*, 30:1212–1229, July 2008.

[83] Hirotake Yamazoe, Akira Utsumi, Tomoko Yonezawa, and Shinji Abe. Remote gaze estimation with a single camera based on facial-feature tracking without special calibration actions. In *ETRA*, pages 245–250, 2008.

[84] Neil Robertson and Ian Reid. Estimating gaze direction from low-resolution faces in video. In *ECCV LNCS*, volume 3952, pages 402–415, 2006.

[85] R. Stiefelhagen, M. Finke, J. Yang, and A. Waibel. From gaze to focus of attention. In *Visual information and information systems*, pages 761–768, 1999.

[86] Karthik Sankaranarayana, Ming-Ching Chang, and Nils Krahnstoever. Tracking gaze direction from far-field surveillance cameras. In *WACV*, pages 519–526, January 2011.

[87] N. Krahnstoever, Ting Yu, Ser-Nam Lim, Kedar Patwardhan, and Peter Tu. Collaborative real-time control of active cameras in large scale surveillance systems. In *ECCV M2SFA2*, 2008.

[88] Ting Yu, Sernam Lim, Kedar Patwardhan, and Nils Krahnstoever. Monitoring, recognizing and discovering social networks. In *CVPR*, pages 1462–1469, 2009.

[89] Ming-Ching Chang, Nils Krahnstoever, Sernam Lim, and Ting Yu. Group level activity recognition in crowded environments across multiple cameras. In *AMMCSS*, pages 56–63, 2010.

[90] N. Gourier, J. Maisonnasse, D. Hall, and J. L. Crowley. Head pose estimation on low resolution images. In *R. Stiefelhagen and J.S. Garofolo (eds.) CLEAR 2006 LNCS*, volume 4122, pages 270–280. Springer-Verlag, 2007.

[91] T. Hoedl, D. Br, U. Soergel, and M. Wiggenhagen. Real-time orientation of a PTZ-camera based on pedestrian detection in video data of wide and complex scenes. In *ISPRS*, pages 663–668, 2008.

[92] P. Ozturk, T. Yamasaki, and K. Aizawa. Tracking of humans and estimation of body/head orientation from top-view single camera for visual focus and attention analysis. In *ICCV Computer Vision Workshop*, pages 1020–1027, 2009.

[93] 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 *PETS*, pages 67Ű–74, 2006.

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

[95] N. Metropolis, A. W. Rosenbluth, N. Rosenbluth, A. H. Teller, and E. Teller. Equation of state calculation by fast computing machines. *JCP*, 21:1087–1092, 1953.

[96] William Brendel and Sinisa Todorovic. Learning spatio temporal graphs of human activities. In *ICCV*, 2011.

[97] Yan Ke, Rahul Sukthankar, and Martial Hebert. Event detection in crowded videos. In *ICCV*, 2007.

[98] Jingen Liu, Mubarak Shah, Benjamin Kuipers, and Silvio Savarese. Cross-view action recognition via view knowledge transfer. In *CVPR*, 2011.

[99] Subhransu Maji, Lubomir Bourdev, and Jitendra Malik. Action Recognition from a Distributed Representation of Pose and Appearance. In *CVPR*, 2011.

[100] U. Gaur, Y. Zhu, B. Song, and A. Roy-Chowdhury. A string of feature graphs model for recognition of complex activities in natural videos. In *ICCV*, 2007.

[101] Suha Kwak, Bohyung Han, and Joon Hee Han. Scenario-based video event recognition by constraint flow. In *CVPR*, 2011.

[102] Vlad I. Morariu and Larry S. Davis. Multi-agent event recognition in structured scenarios. In *CVPR*, 2011.

[103] Du Tran and Junsong Yuan. Optimal spatio-temporal path discovery for video event detection. In *CVPR*, 2011.

[104] Gloria Zen and Elisa Ricci. Earth Movers's Prototypes: a Convex Learning Approach for Discovering Activity Patterns in Dynamic Scenes. In *CVPR*, 2011.

[105] R. Mehran, A. Oyama, and M. Shah. Abnormal crowd behavior detection using social force model. In *CVPR*, 2009.

[106] Jian Li, Shaogang Gong, and Tao Xiang. Scene segmentation for behavior correlation. In *ECCV*, 2008.

[107] Shandong Wu, Brian E. Moore, and Mubarak Shah. Chaotic invariants of lagrangian particle trjectories for anomaly detection in crowded scenes. In *CVPR*, 2010.

[108] Vijay Mahadevan, Weixin Li, Viral Bhalodia, and Nuno Vasconcelos. Anomaly detection in crowded scenes. In *CVPR*, 2010.

[109] Daniel Kuettel, Michael D. Breitenstein, Luc J. Van Gool, and Vittorio Ferrari. What's going on? discovering spatio-temporal dependencies in dynamic scenes. In *CVPR*, 2010.

[110] Timothy Hospedales, Shaogang Gong, and Tao Xiang. A Markov clustering topic model for mining behaviour in video. In *ICCV*, 2009.

[111] Arslan Basharat, Alexei Gritai, and Mubarak Shah. Learning object motion patterns for anomaly detection and improved object detection. In *CVPR*, 2008.

[112] Ming-Ching Chang, Nils Krahnstoever, and Weina Ge. Probabilistic group-level motion analysis and scenario recognition. In *ICCV*, 2011.

[113] T. Yu, Y. Wu, N. Krahnstoever, and P. Tu. Distributed data association and filtering for multiple target tracking. In *CVPR*, 2008.

[114] Gabriella Csurka, Christopher R. Dance, Lixin Fan, Jutta Willamowski, and Cédric Bray. Visual categorization with bags of keypoints. In *In Workshop on Statistical Learning in Computer Vision, ECCV*, 2004.

[115] Angela Yao, Juergen Gall, and Luc Van Gool. A hough transform-based voting framework for action recognition. In *CVPR*, 2010.

[116] Fabian Nater, Helmut Grabner, and Luc Van Gool. Exploiting simple hierarchies for unsupervised human behavior analysis. In *CVPR*, 2010.

[117] Heng Wang, Alexander Kläser, Cordelia Schmid, and Liu Cheng-Lin. Action recognition by dense trajectories. In *CVPR*, 2011.

[118] Imran Saleemi, Khurram Shafique, and Mubarak Shah. Probabilistic modeling of scene dynamics for applications in visual surveillance. *PAMI*, 31:1472–1485, August 2009.

[119] Shobhit Saxena, François Brémond, Monnique Thonnat, and Ruihua Ma. Crowd behavior recognition for video surveillance. In *ACIVS*, 2008.

[120] Ivan Laptev, Marcin Marszalek, Cordelia Schmid, Benjamin Rozenfeld, Inria Rennes, Irisa Inria Grenoble, and Lear Ljk. Learning realistic human actions from movies. In *CVPR*, 2008.

[121] Jingen Liu, Jiebo Luo, and Mubarak Shah. Recognizing realistic actions from videos in the wild. In *CVPR*, 2009.

[122] Lixin Duan, Dong Xu, Ivor W. Tsang, and Jiebo Luo. Visual event recognition in videos by learning from web data. In *CVPR*, 2010.

[123] William Brendel, Sinisa Todorovic, and Alan Fern. Probabilistic event logic for interval-based and holistic event recognition. In *CVPR*, 2011.

[124] Asaad Hakeem and Mubarak Shah. Learning, detection and representation of multi-agent events in videos. *Artifical Intelligence*, 171:586–605, June 2007.

[125] Wongun Choi, Khuram Shahid, and Silvio Savarese. What are they doing?: Collective activity classification using spatio-temporal relationship among people. In *Visual Surveillance Workshop, ICCV*, 2009.

[126] Wongun Choi, Khuram Shahid, and Silvio Savarese. Learning context for collective activity recognition. In *CVPR*, 2011.