Invited Paper

J Inf Process Syst, Vol.10, No.1, pp.1~22, March 2014
http://dx.doi.org/10.3745/JIPS.2014.10.1.001

pISSN 1976-913X
eISSN 2092-805X

Janus - Multi Source Event Detection and Collection System for Effective Surveillance of Criminal Activity

Cyrus Shahabi*, Seon Ho Kim*, Luciano Nocera*, Giorgos Constantinou*, Ying Lu*, Yinghao Cai*, Gérard Medioni*, Ramakant Nevatia*, and Farnoush Banaei-Kashani*

Abstract—Recent technological advances provide the opportunity to use large amounts of multimedia data from a multitude of sensors with different modalities (e.g., video, text) for the detection and characterization of criminal activity. Their integration can compensate for sensor and modality deficiencies by using data from other available sensors and modalities. However, building such an integrated system at the scale of neighborhood and cities is challenging due to the large amount of data to be considered and the need to ensure a short response time to potential criminal activity.

In this paper, we present a system that enables multi-modal data collection at scale and automates the detection of events of interest for the surveillance and reconnaissance of criminal activity. The proposed system showcases novel analytical tools that fuse multimedia data streams to automatically detect and identify specific criminal events and activities. More specifically, the system detects and analyzes series of incidents (an incident is an occurrence or artifact relevant to a criminal activity extracted from a single media stream) in the spatiotemporal domain to extract events (actual instances of criminal events) while cross-referencing multimodal media streams and incidents in time and space to provide a comprehensive view to a human operator while avoiding information overload. We present several case studies that demonstrate how the proposed system can provide law enforcement personnel with forensic and real time tools to identify and track potential criminal activity.

Keywords—Multi-source, Multi-modal Event Detection, Law Enforcement, Criminal Activity, Surveillance, Security, Safety

1. INTRODUCTION

With the increasing instrumentation of our urban centers, law enforcement agencies continue to invest in technologies that promise to provide increased awareness of criminal activity and

※ This research has been funded in part by Award No. 2011-IJCX-K054 from the National Institute of Justice, Office of Justice Programs, U.S. Department of Justice, the USC Integrated Media Systems Center (IMSC) and unrestricted cash gift from Northrop Grumman. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of any of the sponsors such as the Department of Justice.

Manuscript received February 9, 2014; accepted March 2, 2014.

Corresponding Author: Seon Ho Kim (seonkim@usc.edu)

* Integrated Media Systems Center, University of Southern California, Los Angeles, CA, USA ({shahabi, seonkim, nocera, gconstan, ylu720, yinghaoc, medioni, nevatia, banaeika}@usc.edu)

means to automate the surveillance and response to criminal activity.

Existing systems have been developed with a focus on video surveillance data in an attempt to capitalize on deployments of a large number of surveillance cameras. However, recent technological advances have enabled the possibilty to expand the scope of these systems to include not only video data from security cameras, but also data from various other sources such as actuated surveillance cameras, geospatial data (e.g., gazetteers), social media data (e.g., Tweets, public personal information on the Web such as Facebook and Google+ posts), wearable sensors data (e.g., Google Glass), to name a few. Meanwhile, advances in video and text analytics are creating the unique capability to utilize these data for detecting, identifying and responding to criminal events in increasingly instrumented urban areas. Thus, there have been significant interests in the technologies and systems to effectively harness multimodal media streams for surveillance. However, such a system has not yet fully materialized due to the challenges related to the volume and speed at which relevant incoming data are being generated, and due to the current relatively slow analytics capabilities (in particular video and image analytics). While some work has been done to address this need ([1], [2]), to the best of our knowledge, none of the proposed approaches has been comprehensive enough to provide the required level of multi-modal data integration and automation.

We identify the following challenges for effective multi-modal surveillance system: 1) Dealing with large volume of data in detecting events/activities, particularly with events that occur in a large area over a long time period, 2) Difficulty in detecting and characterizing criminal activity from a single data stream (e.g., infering a shooting event only by examining Tweets), and in integrating relevant information across multiple streams (e.g., relating target objects across multiple video feeds solely based on their visual appearance), 3) Timely acquisition and short response time needed to provide relevant information to a task, i.e., acquiring positive identification of potential suspects and providing law enforcement personnel with the ability to timely relate this information to other relevant data.

To overcome these challenges, we propose a framework for multisource event detection and collection system, named *Janus*, for effective surveillance of criminal activity utilizing content analysis and spatiotemporal database techniques that can provide law enforcement with the means to effectively respond to criminal activity at scale. More specifically, we rely on the hypothesis that *multi-source and multi-modal integration enables more effective surveillance systems by leveraging complementary properties of the data collected from multiple and various sources*. Our main approach is to exploit the spatiotemporal relation among multisource media contents. At its core, the proposed system utilizes state-of-the-art content analysis techniques to detect *incidents* (we define an incident as data extracted from a single media stream that can be useful to understand and characterize criminal activity) from textual and video data streams, while integrating them in the spatio-temporal domain (rather than content domain) to identify *events* (we define an event as a possible instance of a specific criminal activity). The key justification behind using content for incident detection and spatiotemporal features for cross-referencing is that while computers can extract incidents from individual data streams quite efficiently, relating these incidents based on their contents has proven to be hard, particularly in the presence of a large dataset. We show that our approach is versatile and scalable with respect to the volume and type of data streams. We further demonstrate a prototype implementation of the proposed system with case studies showing how such a system could be effectively used for both real time surveillance and the forensic analysis of criminal activity.

The remainder of this paper is organized as follows. In Section 2, we describe background related to our approach. Section 3 presents the system architecture, describes the data types considered, and provides the details of the most important components of the system. Implementation details of the proposed system are further discussed in Section 4 with case studies. Section 5 concludes the paper and discusses the future directions.

2. BACKGROUND

The indexing and integration of video streams with other sources of information have been extensively studied in the last two decades. Some studies suggest the utilization of both audio and video input for event detection and indexing [1]. Other work extends these ideas to include features extracted from the video itself, such as text [2] while some studies suggest the indexing and integration of multiple video streams: these can be sourced from the same modality camera, (e.g., visible light cameras) [3], [4] or from cameras of different modalities (e.g., infrared and visible light) [5], [6]. A comprehensive review of these methods in the context of surveillance can be found in [7]: importantly all these studies utilize data that are extracted solely from video sources, i.e., none of these studies explicitly address the task of combining sources of different modalities for event detection.

Combining image/video data with its corresponding time and location can provide an effective way to index and search videos, especially when a database handles an extensive amount of data in a scalable system. There have been significant researches on organizing and browsing photos according to location and time. A number of studies ([8], [9], [10]) have introduced a metadata powered image search and/or built a database, which indexes photographs using time and location coordinates (latitude/longitude from GPS). All these techniques use only the camera locations as the reference (i.e., a point in geo-space) in describing images. This point representation may not be sufficient to support indexing and searching of geo-tagged videos at the high semantic level as needed by humans. Humans are interested in the object itself and its location, not the camera location, and there can be discrepancy between the camera position and the location of object the scene shows.

Studies that do try to infer the location of the observed scene through GPS, text and visual data are geared at recovering the 3D structure of the scene ([11], [12]). These algorithms usually rely on large photo collections – either publicly available or collected by a specific user community - rather than surveillance videos. The extensive computational power needed to complete these tasks does not allow for the extension of these algorithms to real-time or even near-real-time video applications.

Conventional ways to organize and index videos are similar to the ones used for images in the sense that a video can be represented as a camera point or a trajectory representing the camera movement. Recent advances in sensor technologies allow video clips to be tagged with geographic properties (e.g., camera direction obtained using a digital compass and the camera location from GPS) while being collected. Such metadata can be attached to the video streams automatically, hence allowing for consistent annotation of the collected video contents. This meta-data enables using various criteria for versatile video search. The captured geographic metadata have significant potential for aiding the indexing and search of geo-tagged video data.

However, there has been little research on utilizing such metadata for the systematic indexing and search of video data.

In this paper we extend prior studies on video analysis ([13], [14], [15]) and spatiotemporal indexing ([16], [17], [18]) to enable compact and efficient registration of incidents and events and consequently index them in a spatiotemporal database to enable more efficient criminal activity surveillance.

3. SYSTEM

In this section, we provide an overview of the proposed system architecture and briefly explain the system modules and data types.

3.1 Overview

As shown in Fig. 1, our proposed system follows a three-tier architecture comprising data, analytics, and presentation tier. Each tier processes incoming data streams, and can produce new streams and/or make streams or historical data available to the other tiers. Ultimately the extracted streams of incidents, events and raw data streams, both incoming and stored are made available to end users at the presentation level.

Data Tier The data tier is the bottom tier that receives the incoming raw or pre-processed data from available data stream as the system inputs. Supported data streams are produced by remote data sources including structured text (i.e., incident reports, ACR (access control readers) readings, unstructured texts (i.e., Twitter messages), mobile videos (i.e., crowdsourced videos) and videos from regular surveillance cameras and Pan-Tilt-Zoom (PTZ) cameras. *Incident Extraction Module* analyzes the received data and detects predefined incidents. *Data Management Module* that is charge of managing storage and access to incoming data and incidents.

Analytics Tier The analytics tier, also called the *middle tier*, provides analytics and data query capabilities. This tier includes the *Event Detection Module* that extracts events from the incidents detected in the Data Tier. The *Space-Time Cross Referencing Module* cross-references raw streams, incidents, and events in spatial and time domain. The *Data Query Web Services* provides a standardized interface to all available data streams.

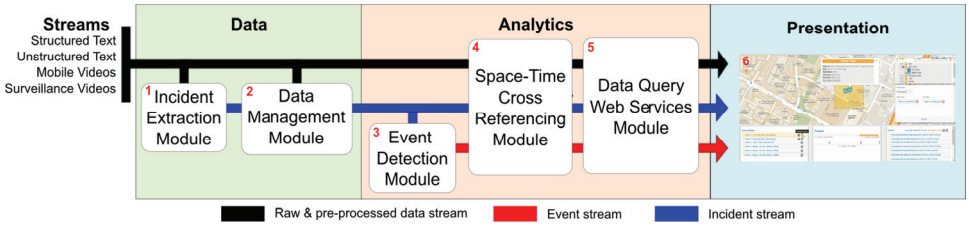


Fig. 1. System Architecture: Data, Analytics and Presentation tiers are shown in green, orange and blue. Color-coded arrows outline how raw and pre-processed data, incidents and events are routed through the modules of the system

Presentation Tier The main functionality of this tier, also called the *top tier*, is to provide end-users with search, query, and filtering capability and to present the results to the user through an interactive web-enabled interface. Raw and pre-processed stored and streaming data, incidents and events data can all be queried and visualized in context in this presentation layer.

Data Sets In our implementation of the system we have considered a comprehensive range of datasets including: 1) ACD (Access Control Data), i.e., sensor readings generated by card/badge readers installed at a number of the entrances to the buildings, 2) LPR (License Plate Readings), i.e., zoom-in images as well as OCR/Text conversion of the license plates of the vehicles captured by special license plate recognition cameras, 3) internal incident reports, i.e., reports developed and maintained by law enforcement, 4) public incident reports, i.e., reports of the incidents that law enforcement announces on its publicly accessible website, 5) Tweets, i.e., relevant tweets collected from the publicly available twitter messages, 6) videos collected by PTZ Network Cameras that capture high-resolution facial images, 7) trajectories of pedestrians extracted from PTZ and regular surveillance cameras, and 8) video data collected by mobile devices (such as voluntarily contributed or crowdsourced smartphone videos).

Mobile video dataset is a video dataset recorded in casual way (e.g., street shot). The rationale for studying mobile video in our project is that sometimes we can detect some events based on this kind of video dataset. For example, we can find crime clues from the video taken by a passerby in the same time and location of the crime. In our project, we support spatial temporal queries on mobile videos. The new features of our project distinguishing from existing applications (such as Youtube, Klip, Keek, etc.) are following: 1) We treat a video as a set of video frames. Our spatio-temporal queries are based on the video frames instead of the entire video. However, in existing application, each video is treated as one entity, whereas it consists of continuous frames. For a video with long duration time, user may have interest for some specific frames in certain time range. In our project, we split the video into video frames so that we can return users their specified spatial temporal queries exactly. 2) In our system, we make full use of the video metadata, which are automatically captured from GPS and compass sensor in smartphones, to process users' spatial temporal queries [18]. To sum up, each data stream is processed by module 1 in Fig. 1.

3.2 System Modules

In this subsection, we explain the details of the basic system modules in Fig. 1.

3.2.1 Incident Extraction Module

An incident is the data extracted from a single media stream that can be useful to understand and characterize criminal activity. Incident extraction method is data dependent and realized by module 1 of Fig. 1. More specifically, the Incident Extraction Module instantiates specific data processing components that produce artifacts, which in turn are used to identify and extract incidents. For instance, to detect a following event (when a person is following another person, see details in Section 4.4), we integrate following incidents detected on single video streams. Following incident results from the analysis of extracted trajectories of pedestrians in a video stream; while the following event is inferred by relating those following incidents (and other pertinent incidents) across video streams that involve the same persons. In the following, we

detail how PTZ Face Detection and Tracking and Multiple Target Tracking in video data produce artifacts (high-resolution face images, trajectories, and appearance of pedestrians) that can be used to infer following incident and events. Moreover, we detail how the analysis of Textual Incident Data and Tweets produces another source of incidents.

PTZ Face Detection and Tracking

Having high-resolution face imagery is very useful for both real-time and forensic applications as the facial image helps identify the target of interest. However, regular surveillance cameras only capture a small number of pixels on the face region, resulting in difficulty of using face images. PTZ cameras which can pan, tilt and zoom are powerful tools in far-field scenarios since zooming offers the option to capture a close view at a high resolution on demand. In our system, both the captured high-resolution face images and the extracted trajectories are considered as incidents, the availability of which opens up new opportunities for further biometric analysis and persistent surveillance.

More specifically, the system first detects and tracks every pedestrian entering the field of view of the camera in zoomed-out mode, then selects, using a scheduler, a person to zoom in. After zoom in, the system returns to the wide area mode, and resolves the person-to-person, face-to-person and face-to-face associations. The output that results in incidents streams is a set of geo-tagged and time-stamped high-resolution facial images and time-stamped geo-locations of the pedestrian trajectories. The overview of PTZ face detection and tracking system is shown in Fig. 2 (a). There are two modes in the system, the zoomed-out mode, Fig. 2 (b), and the zoomed-in mode, Fig. 2 (c). The processes of multiple targets tracking in zoomed-out stage, face detection and association in zoomed-in stage and output results are illustrated in Fig. 3 where pedestrians' trajectories and the associated high-resolution faces extracted are marked with the same color. For a more in depth account of the underlying methods and algorithms refer to [19].

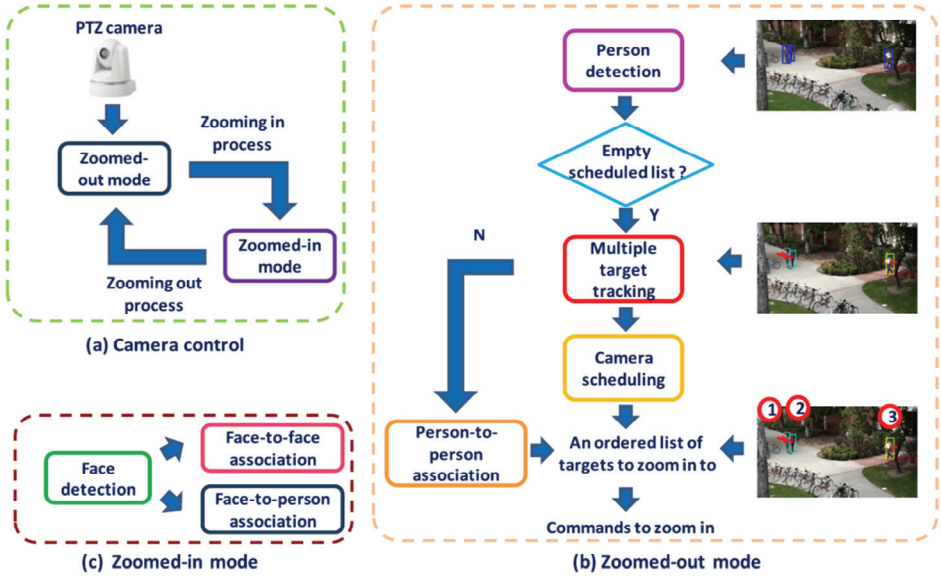
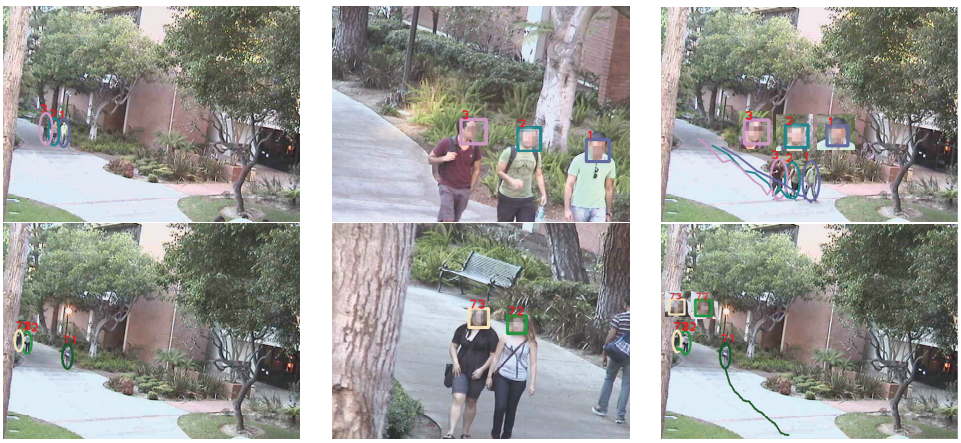


Fig. 2. Overview of PTZ face detection and tracking system



(a) (b) (c)

Fig. 3. Annotated PTZ face detection and tracking system scenes. (a) Zoomed-out stage. (b) Zoomed-in stage. (c) Associated high-resolution faces and trajectories. For privacy concerns, actual high-resolution extracted face are shown blurred

Multiple Targets Tracking Within and Across Multiple Cameras

The trajectory, which records the target's location from entry to exit of a scene is the basic building block needed to detect incidents from video sequences. For example, incidents such as "following", "illegal entry", and "meeting" can easily be inferred from trajectories of people in video sequences. Multiple target tracking algorithms which assign consistent labels to objects in different frames of a video are used to obtain trajectories of objects. Many approaches for object tracking have been proposed in recent years.

However, the field of view of a single camera is limited; multiple cameras provide a solution to wide area surveillance by extending the field of view of a single camera. Related challenges pertain on how to track multiple targets within camera and how to associate trajectories across cameras. In this section, we discuss our approach to intra-camera and inter-camera multiple target tracking.

Intra-camera Multiple Target Tracking Most previous methods of multiple target tracking can be classified into Association Based Tracking (ABT) and Category Free Tracking (CFT). ABT associates detection responses into tracks, while CFT tracks a manually tagged region without object category information or pre-trained detectors. Standard ABT approaches perform well on objects that can be well detected by an offline-trained detector, like pedestrians [20], vehicles, etc. However, most real surveillance data contains human activity with articulated pose changes, during which large object appearance deformation exists and detection results become unreliable, in which case ABT might produce incomplete tracks or linking errors. Therefore, we aim at incorporating merits of both ABT and CFT in a unified framework. In our approach, the standard ABT tracking method [21][22] is first applied to an input video to generate pedestrian tracking results. Then, we build a specific appearance model for each tracklet by collecting positive and negative training samples. Details about how to learn a superpixel based appearance model can be found in [21]. After initial training and obtaining an appearance model, we follow

the CFT technique to grow initial tracklets by exploring articulated pose changes or heavy occlusions. We further allow two additional modules including pose transition constraints and articulated human detectors. Fig. 4 shows some sample results of intra-camera multiple target tracking.

Inter-camera Multiple Target Tracking We assume that the trajectories within cameras are already obtained in the intra-camera tracking stage. In addition, we assume the spatial connectivity between entry zones and exit zones in multiple cameras is learned. The aim of inter-camera multiple target tracking is to associate the trajectories of people when they move from one camera's field of view to another.

If only snapshots of people from multiple cameras instead of tracks are available, matching snapshots across cameras is usually termed “person re-identification”. Without context information from videos, even human observers experience difficulties to tell people apart. Here, we mainly explore what kind of context information from videos can be used for inter-camera tracking. We introduce two kinds of context information, spatio-temporal context and relative appearance context.

The spatio-temporal context indicates a way of collecting samples for discriminative appearance learning where target-specific appearance models are learned to distinguish different people from each other. The relative appearance context models inter-object appearance similarities for people walking in proximity to each other. The relative appearance model helps disambiguate individual appearance matches across cameras. (a) Camera 1 (b) Camera 2 Fig. 5 shows some sample results of inter-camera tracking. People traveling between cameras are linked with solid lines. It shows that our method finds correct associations between targets even in a crowded scene.

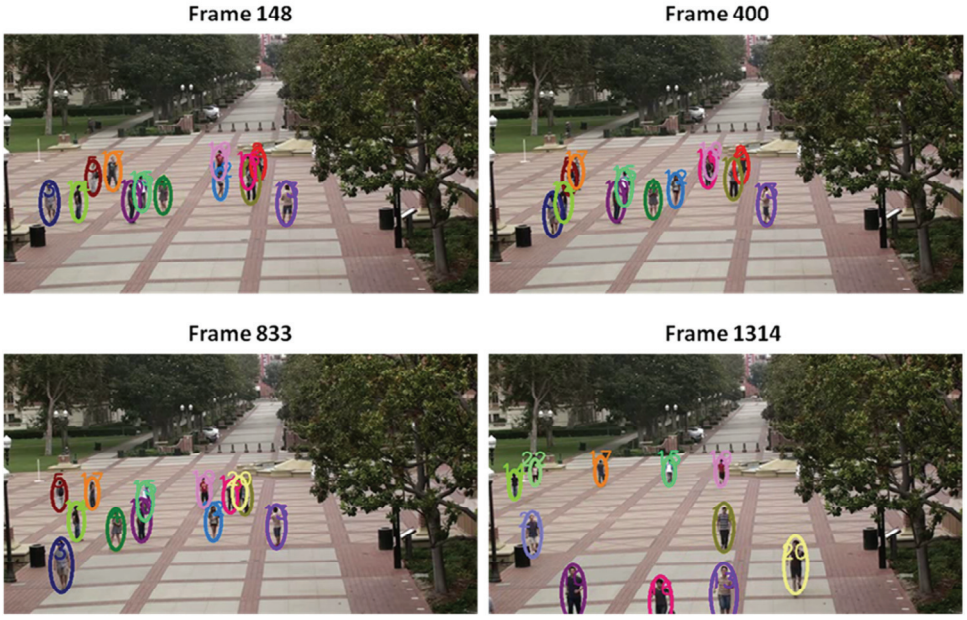
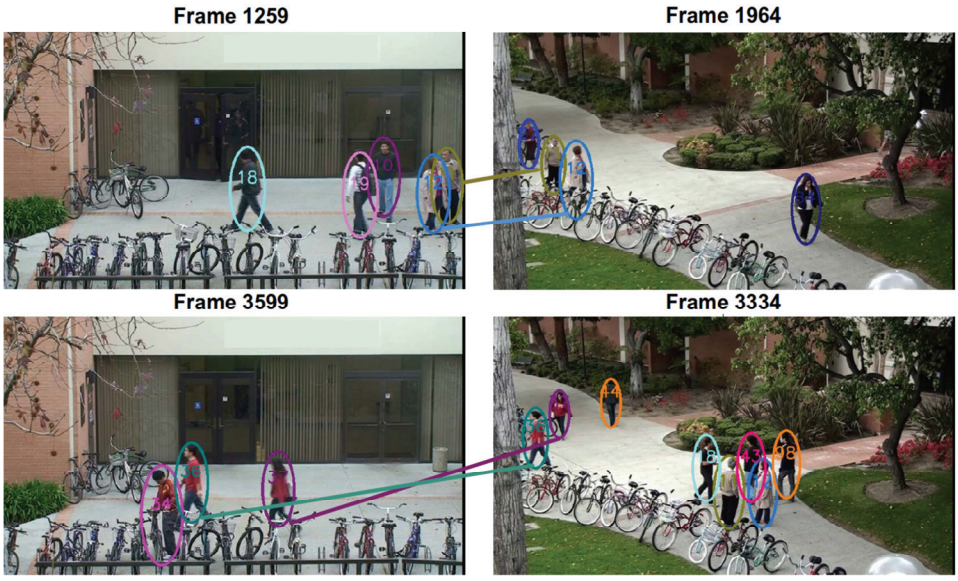


Fig. 4. Results of intra-camera multiple target tracking. Tracked pedestrians are shown with the same color and id



(a) Camera 1 (b) Camera 2

Fig. 5. Results of inter-camera multiple target tracking

Analysis of Textual Incident Reports

This module analyzes texts to identify the following attributes – actors, victims and objects from the incident report summary published by law enforcement on a daily basis. An actor is a person who is involved in a criminal activity. A victim is a person who has suffered due to the actor. Objects are things lost, broken, stolen in the activity. For example, the summary field in a report could be: "A student reported his laptop and tablet computers missing". Hence, the victim is 'student' and objects are 'laptop' and 'tablet computers'.

The module utilizes the Annotation Query Language (AQL) which is available in IBM InfoSphere Streams. Regular expressions are used in AQL to extract snippets such as summary from textual report data. A simple pattern example to extract actors ('assaulted by a coworker') is the following:

(<V.verb>)'by'(<D.determiner>)(<N.noun>|<A.adjective)

Analysis of Tweets

This module helps in identifying the entities – culprit and victims from tweets for following incidents happened in and around the campus. The process works as follows: 1) IBM InfoSphere Streams is connected to Twitter Streaming API by using the HTTP Utility toolkit. We fetch tweets in JSON format that contain the keyword "stalk". 2) Extracted tweets are filtered out (re-tweets, use of friendly abbreviations etc) and their content is processed with AQL to identify victims and culprits. A simple pattern example to extract culprit (matches 'stalked by Darcey') is the following:

('stalked'|'Stalked')'by'<D.determiner>?(<A.adjective>?<N.noun>{1,3})

Finally, 3) the extracted results are stored in our database.

3.2.2 Data Management Module

The Data Management Module manages incoming raw data streams and provides storage and pre-processing. In the following paragraph we highlight the most important functionality for this module.

Video Segmentation and Storage

Since video cameras can produce large amounts of data over time, to limit the storage space needed for the raw video data, we implement segmentation methods to discard those segments where nothing of interest happens, and only store the segments that are informative to capture incidents. More specifically, we detect motion in live video stream and only store those segments where motion is present. We have implemented video segmentation on IBM InfoSphere Streams. In InfoSphere Streams, continuous applications are composed of individual operators, which interconnect and operate on multiple data streams. Data streams can come from outside the system or be produced internally as part of the application. More specifically, we route the video signal via the Real Time Streaming Protocol (RTSP) and utilize the InfoSphere Streams to process the video data to segment in real time based on the presence of motion components. Fig. 6 shows the architecture of our implementation. Processed output segments are stored in the database server. This process is repeated while the RTSP link is active.

3.2.3 Event Detection Module

The Event Detection Module is responsible for extracting potential criminal activity events, which can be done by integrating incidents. We defer the description of how this is carried out to Section 4.4 in the case of the forensic analysis of following, Section 4.5 in the case of real-time analysis of following and Section 4.6 in the case of real-time analysis of shooting.

3.2.4 Space-Time Cross-Referencing and Data Query Web Services Modules

The Space-Time Cross Referencing Module provides spatial and temporal cross-referencing of raw input streams, incident streams and event streams. There is a large amount of location-based and time-based information generated and used by Janus. Examples of such data include crime incidents, video feeds and tweets. Inevitably, Janus is expected to enable users to search not only by keywords but also by specifying the locations and times associated with their desired objects.

To enable this capability we need to define a new scoring mechanism to compute the spatial-temporal-textual (STT) relevance of an object to a query. An object can be any unit of data with

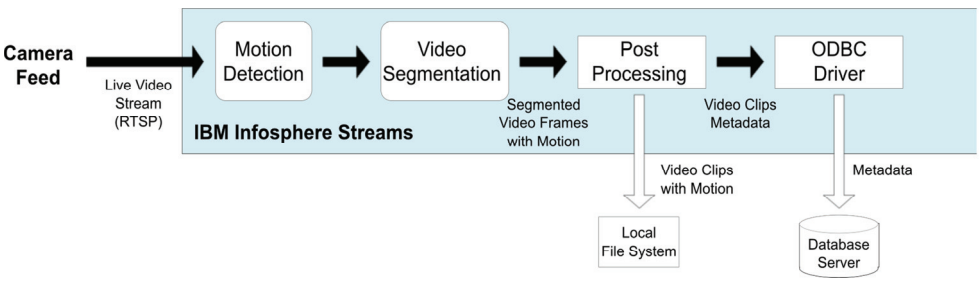


Fig. 6. Video storage system architecture

textual, spatial and temporal features and we wish to develop a general enough approach that can be applied to any object with textual, spatial and temporal features. We devised the STT ranking into two parts spatial-textual relevance ranking and temporal-textual relevance ranking.

With spatial-textual ranking, the focus is on ranking objects by combining spatial and of incidents features and the query. With temporal-textual ranking, the focus is on integrating the temporal dimension of the objects as well as the query to the search and ranking process. The goal of our temporal-textual ranking is to identify objects that are both temporally and textually relevant to the query and rank them based on the combined temporal-textual relevance. Since the proposed models for both spatial-textual and temporal-textual are similar and share many common characteristics, we can easily combine the two models and develop a new model for a seamless spatial-temporal-textual relevance ranking.

We have developed this capability by leveraging our recent work in which we addressed the problem of ranking web documents based on their spatial and textual features [23]. In particular, we first proposed a ranking method that in order to compute the relevance score of a document for a query, considers not only the frequencies of the query keywords in a document but also the spatial overlap of the document with the query. We also developed a new hybrid index called spatial keyword inverted file (SKIF) and two algorithms for the efficient processing of spatial-keyword queries. We also extended our research to cases where query and document locations are geographical points (versus spatial regions) [24]. In a more recent paper [25], we also proposed a new ranking algorithm for temporal-textual search and queries. Built on top of above three studies, where we define a new scoring mechanism to compute the STT relevance of an object to a query. Similar to [23] and [25], the idea is to follow the same intuitions and concepts used in regular (textual) searches by defining the corresponding concepts and parameters for spatial and temporal data. The developed similarity measure captures all STT features of the objects while allowing for tunable weights for different features. In order to do this, we devised a ranking method that in order to compute the relevance score of an object for a query, considers not only the textual feature of each object (textual keywords) but also the spatial overlap (or proximity) of the object with (spatial part of) the query and temporal overlap (or proximity) of the object with (temporal part of) the query. As a result three relevance scores between the query and each object are generated: textual relevance, temporal relevance and spatial relevance. Each relevance score then gets normalized (divided by the max possible score for that relevance score). Finally, three relevance scores are aggregated using three tunable weights x, y and z. These weights are determined by the users and in all cases $x + y + z = 1$. These weights also let users to query the objects based on only 1 or 2 dimensions. For instance, users can search and query only for objects with certain keywords or they can search based on the combination of space and time and ask the system to rank objects based on spatio-temporal features of the objects. Our ranking model supports both point-based and regions based scenarios. In other words, it works well whether spatial/temporal features of objects (and query) are points or regions (e.g., geographical points, time instances or spatial regions and time intervals). We use proximity (in space or time) for point-based queries and spatial/temporal overlap for region-based spatial/temporal queries. In summary, In summary, using the techniques just described we have provide users of the system the capability to perform STT search and query: the user specifies keywords as text strings, location as regions/points on Google Map and time as temporal interval/instance. In return, it generates a ranked list of the objects that satisfy the textual, spatial and temporal constraints specified by the query; the result-set is displayed on

Google Map as well as a textual list to the querying user.

3.2.5 Data Query Web Services Module

All data streams and stored data are finally made available to the application level through dedicated web services.

4. IMPLEMENTATION / CASE STUDIES

The system runs on top of Windows Server 2008 OS. We have used Tomcat as our application server which hosts the HTML pages and our web services implemented as servlets. On the back end, the data are stored in Oracle DB server that supports spatial queries. For the purpose of testing we use off-the-shelf Sony PTZ Network Camera SNC-RZ50N to capture face images and Pelco PTZ Network cameras as well as Sony digital video recorders to extract people's trajectories. Tens of regular surveillance cameras around USC campus, operated by the Department of Public Safety, were used to collect input video streams. Mobile videos from smartphones were also collected for the case studies.

4.1 User Interface

To provide the end users with a graphical front end accessible on desktop and mobile devices we have opted to implement the system's GUI as a fully integrated web-enabled application using state of the art web technologies, including HTML5, CSS3, JavaScript, jQuery, jQueryUI and Google Maps API. The User Interface consists of six modules as depicted in Fig. 7: a) the map where all results are displayed b) the query history where previous queries are listed, c) the timeline which can be used to filter out irrelevant event or/and highlight events based on the temporal attributes of the data, d) the datasets component where the user can choose which datasets will be queried, e) the query formulation component which includes parameters to be used for the spatiotemporal query and f) the event detection panel where detected events are

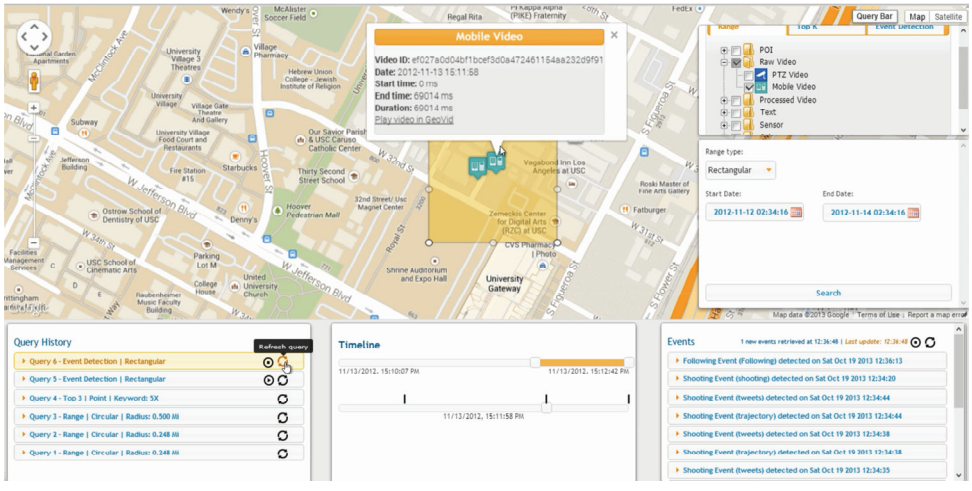


Fig. 7. An exemplary screenshot of user interface

popped up along with the information that generated the event.

The datasets are implemented in such way that the system is easily extensible with additional data sources. Specifically, we grouped the datasets in four categories based on the type of the location and the content of each dataset, i.e., static location and content (e.g., POIs), static location and dynamic content (e.g., PTZ cameras), dynamic location and static content (e.g., trajectories), and finally dynamic location and dynamic content (e.g., mobile videos). According to the category of a dataset, we handle data homogeneously when they are displayed on the map.

The system supports spatiotemporal queries. The spatial region can be selected by three different ways: by defining a) a circular region (single location, i.e., point, and radius), b) a rectangular region (locations of upper left and lower right corners), or c) a trajectory (a list of locations and a radius).

In Sections 4.2 and 4.3 we will briefly review how we have incorporated replay capability and continuous queries, both of which were essential for the practical implementation of the case studies that are presented in Section 4.4, 4.5 and 4.6.

4.2 Real Time Replay

To support the development and demonstration of our real-time event detection case studies we had to resort to simulating real-time data streaming relevant to criminal activity because 1) some of the data types are not always readily available, and 2) the data collected in real-time is not always useful for system performance evaluation and demonstration purposes. To emulate real-time data collection we have therefore developed a “replay” capability that uses previously collected data to simulate incoming data streams. More specifically we have implemented data replay system modules for each one of the data types listed in section 3, and a master replay module that utilizes each data replay module to provide the system with the relevant synchronized raw data streams. Note that only the raw streaming in input is simulated the system is actually running, as it would with live real-time data. We have used this capability for the sake of implementing and demonstrating case studies, however it can be used also as input to a training or assessment application to replay scenarios of interest to law enforcement personnel.

4.3 Continuous Queries

Except from supporting one-time queries on historic data in support of forensic analysis, the system also supports continuous queries. With continuous queries the query is issued once but is continually re-evaluated to reflect updates in the result of the query as new data arrives in real-time. Continuous queries allow for real-time monitoring of a potential criminal situation, and to support automated real-time event detection.

For example, consider a query on LPR data. The result will be a set of LPR data plotted on a map in the specified area and time period of the query, similar to the result shown on the map of Fig. 7. If new LPR data are inserted to the DB from the external source and the query was marked as a continuous query, then the query, periodically, will retrieve these new LPR readings and update the existing map.

Building on top of continuous queries, we developed algorithms for activity recognition based on data analysis. These algorithms extend the classic association rule mining techniques to apply to human location data; hence, spatial association rule mining. With these algorithms, we focus on the case of “following” and “shooting” event detection. In the case of “shooting” event

detection we combine findings from text data such as Tweets.

4.4 Forensic Analysis for Case of Following

Initially, Janus was implemented to support forensic analysis based only on historic data. The incidents collected from various sources are cross-referenced in space and time to present a holistic view to a human operator. The operator, based on the information presented to him, can make and validate hypothesis to ultimately assess and investigate the potential criminal activity. In order to allow for forensic investigation, incidents are indexed by their associated time and space and saved in our database.

For this case study the scenario covers the investigative needs to respond to a report of following made to a police officer. A female student reported that someone was following her on the way back to her on campus apartment. The complainant provides some pieces of information, e.g., path she was walking, date and time of the incident, characteristics of the suspect, car license plates if available, etc.

4.4.1 Implementation

The system supports top-k queries as discussed in section 3.2.4. The user can use the UI to perform the query by setting the date to the reported date, the location, a keyword such as “stalk” and a value for k, e.g. 10. The user then selects the queried sources. Once the query is issued on incident reports (both internal and public) and tweets the top 10 more relevant results (based on location, time and keyword) from each dataset are returned and shown on the map, while the timeline is updated. The officer can go through the results to examine any previously reported “stalking” incidents. In addition, if the complainant was able to capture partial information, such as license plates, the system is able to use this partial information and retrieve relevant data from LPR dataset.

With the range query, the officer can select the area of interest and retrieve the locations of PTZ cameras around the areas. Live video can help better understand the layout of the area where the incident was reported. Also datasets extracted from PTZ videos (i.e. trajectories, face images, stored videos) can be queried and examined when the PTZ cameras could capture the trajectories and high-resolution face images – information that can be readily used to identify the complainant and suspect. Videos captured by passers-by using a mobile device and uploaded can provide additional information.

Finally, since campus dormitories are equipped with ACD sensors, the officer can query ACD records to validate and identify the persons who entered the building as the ACD readings include unique identifiers for specific individuals.

The query history component and timeline are fundamental in this data investigation. The officer can disable previous queries that do not provide any valuable information and enable other queries to isolate “interesting” findings.

4.4.2 Experimental Setup and Results

This scenario demonstrates how the system can enable a human user to effectively explore several geospatial incidents and “connect the dots” for informed decision-making. In our experiments, videos for incident extraction (faces and trajectories) are recorded by three PTZ cameras which are widely separated. The length of the recorded videos is about one hour. Fig. 8

Fig. 8. Example data that can be used for forensic analysis of following: (a) tagged pedestrian tr ajectories, (b) mobile video contributed by the public and (c) automatically extracted traje ctories and high-resolution face images

shows (a) a sample result of multiple targets tracking across two cameras, (b) a mobile video along with its field of view and (c) the extracted trajectories and high-resolution face images from PTZ cameras. By cross-referencing multiple heterogeneous data and providing the tools for querying, visualizing and analyzing the data streams, the complexity of the decision making problem is greatly simplified.

4.5 Real Time Analysis for Case of Following

Following are the one type of potential criminal event that can lead to possible cases of stalking. We detect following events from three data sources: Twitter, Textual Incident Reports and Surveillance Videos. By spatial-temporal cross-referring the event results from the three sources, we can get quick, detailed and accurate following events. For the former two data sources, we detect the following events by analyzing the texts with keywords such as "following" and "followed". For the surveillance videos, we detect following events from trajectories derived from the videos discussed below.

4.5.1 Implementation

Following event detection from trajectories is a non-trivial problem. For example, two very similar or identical trajectories are more likely from two friends walking together than from a following event; similarly two trajectories may potentially constitute a case of following although they are quite different. Thus following trajectories can have various complex scenarios. In this paper, we model these various scenarios using the following four patterns where B is following A: 1) A and B keep a certain distance within which A can be seen by B (denoted as Sight Distance, e.g. in the range [10, 50] meters); 2) A and B have the same speed, direction and distance between each other; 3) B follows A who changes its moving direction suddenly; 4) B follows A who changes its speed suddenly, i.e., when A slows down suddenly, B goes away a little first, then returns and continue to follow. A schematic representations of these four basic patterns in presented in Fig. 9.

We denote $(B_i \rightarrow A_i)$ as Trajectory Fragment Pair (TFP) of A_i and B_i in a time unit, and denote $S(B_i \rightarrow A_i)$ as the *support or possibility* that B_i follows A_i. For Pattern 1, we use distance in equation (1) to formalize *support*, where $avgDist(A_i, B_i)$ is the average distance between A_i and B_i, ε_{imin}, ε_{imax} are the user specified minimum and maximum of Sight Distances. For Pattern 2, we use the velocity difference to formulize the support in equation (2), where $EJ(V_{Ai}, V_{Bi})$ is the

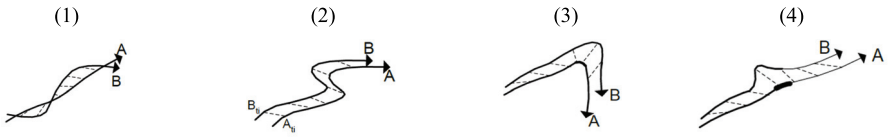


Fig. 9. Four Basic Following Patterns: Arrows indicate the moving direction. Bolder line indicates slower speed

Extended Jaccard (which can consider the size and direction of the velocity together) between Velocities of A_i and B_i, ε_v is the specified velocity difference threshold. Similarly, we use the difference of velocity changes to formalize Pattern 3 and 4, where $EJ(\Delta V_{Ai}, \Delta V_{Bi})$ is the Extended Jaccard between Change of Velocities of A_i and B_i, $\varepsilon_{\Delta v}$ is the specified threshold. Weighting the three measurements together, the formula of the support ($B_i \rightarrow A_i$) is given in equation (4). Inspired by the Association Rule Mining algorithms [26], we update the support of each pair of trajectories by rolling the slide window. If $S(B \rightarrow A)$ is larger than a threshold Ψ, then we report that B follows A.

$$Dist(A_i, B_i) = \left\{ \begin{matrix} 1, & if\ avgDist(A_i, B_i) \in [\varepsilon_{i\min}, \varepsilon_{i\max}] \\ 0, & otherwise \end{matrix} \right\} \quad (1)$$

$$Vel(A_i, B_i) = \left\{ \begin{matrix} 1, & if\ EJ(V_{A_i}, V_{B_i}) > \varepsilon_v \\ 0, & otherwise \end{matrix} \right\} \quad (2)$$

$$CVel(A_i, B_i) = \left\{ \begin{matrix} 1, & if\ EJ(\Delta V_{A_i}, \Delta V_{B_i}) > \varepsilon_{\Delta v} \\ 0, & otherwise \end{matrix} \right\} \quad (3)$$

$$S(B_i \rightarrow A_i) = \frac{W_1 * Dist(A_i, B_i) + W_2 * Vel(A_i, B_i) + W_3 * VelC(A_i, B_i)}{W_1 + W_2 + W_3} \quad (4)$$

4.5.2 Experimental Setup and Results

Textual Incident Report data are collected from law enforcement, and average around 10 reports per day. Video data consisted of 4 videos, about 3 hours long, covering a path around 300 meters long on the USC campus. Ten pairs of volunteers enacted following scenarios thereby generating 10 ground truth cases of following.

A total of 93 trajectories were extracted from the 4 videos and our algorithm detected 12 cases of following. Of these, 9 cases were from the ground truth cases. Therefore, the recall of our following event detection algorithm is 0.9, and the precision is 0.75. Fig. 10 illustrates two detected following events. Specifically, in Fig. 10, a pair of persons (A, B) is detected as following events. B is following A.

4.6 Real Time Analysis for Case of Shooting

Another criminal activity of interest is shooting. As shown in Fig. 11 we emulate real time data using the replay module as discussed earlier in this section. Our event detection is fully customizable. For example, users can define which datasets to be used in the event detection engine, a specific area to be inspected, the way that the data are aggregated over time and space, the sliding window in which the data are analyzed to extract useful features, etc.

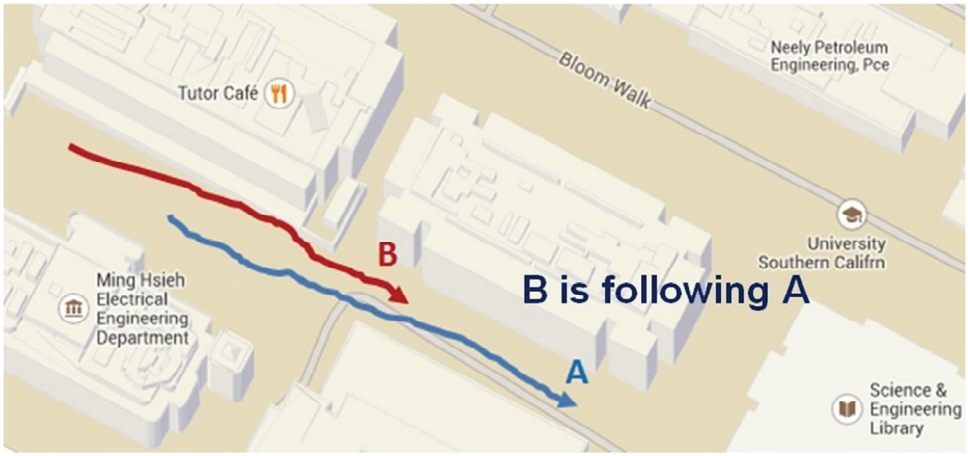


Fig. 10. Illustrates a following results. A pair of persons (A, B) is detected as following events

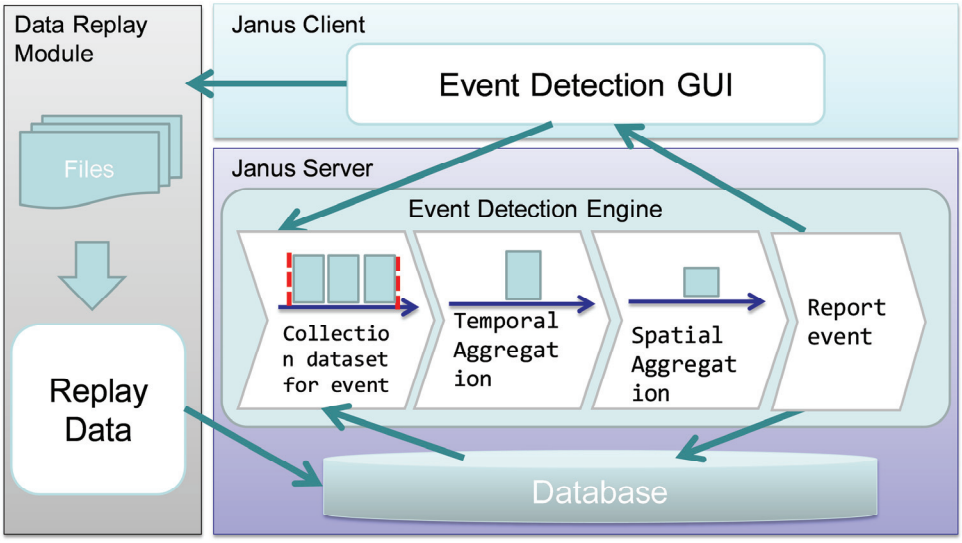


Fig. 11. Shooting event detection

4.6.1 Implementation

For the specific case of shooting the datasets used are predefined; we use tweets gathered from the Twitter API and trajectories of people from PTZ cameras. Given a region where events are going to be detected, we split it into rectangular cells each having equal size.

When using the trajectory data, for each cell we calculate the number of people running (given a speed threshold that determines the average running speed) and we normalize it with the total number of people present in the entire grid. We then apply the MAX aggregation function in a rolling time window given by equation (5), which returns the MAX SLOPE for

each cell between three grids (window size is three). The algorithm tries to identify which cell had a significant change to the number of the running people within a small period of time. This behavior is common when a shooting occurs. We then apply spatial aggregation to the resulting temporal aggregation by considering neighboring cells given by equation (6), which counts the number of neighboring cells that exceed a specific threshold TR. The resulting grid is spatially and temporally aggregated. On this final grid we use another threshold that determines if the suspicion of shooting event is high enough. Similar functionality is used on the tweets datasets.

$$f_{T_{i,j}} = Max(\forall x > y, x = [1,3], y = [1,3], \frac{\left| T_{x_{i,j}} - T_{y_{i,j}} \right|}{t_x - t_y} \tag{5}$$

$$f_{S_{i,j}} = Count(\forall k = [-1,1], A_{T_{(i-k)(j-k)}} > TR) \tag{6}$$

We isolate areas where high volume of tweets related on shooting took place suddenly. Then if both trajectories and tweets reported a shooting event, the likelihood of shooting is increased and the user is notified. Note that in this scenario, each dataset can report a shooting event separately and/or in combination.

The system allows the user to define the functions to be applied in both the spatial and temporal aggregations. For the threshold *TR*, lower values can increase the "sensitivity" of the system while higher values can miss many events. Threshold *TR* is appropriately set in our system. Adaptive threshold parameter learning is deferred for future work.

4.6.2 Experimental Setup and Results

To simulate the shooting event, 15 student volunteers acted in a controlled area. We recorded the action using one installed video recorder in the Tommy Trojan Square on the USC campus. The length of the recorded video is about half an hour. The students walked normally and then started to run at the same time away from the area. Fig. 4 shows some sample results of multiple target tracking. For the tweets we used the replay functionality. The system was able to report a shooting event on the area along with the distribution of people at real time.

These simple case study examples show how the system can use the underlying spatiotemporal features of data and how they can be cross-referenced to detect suspicious events.

5. CONCLUSION

In this paper we described the design and implementation of a multi-source event detection system (Janus) for the effective surveillance of criminal activity and its underlying components. We explained how different datasets can be used and combined using their spatiotemporal features and provided the case studies of real-time criminal activity detection with event detection for a case of following and shooting.

Part of our future work is to develop and validate robust metrics for a wider range of criminal activities. Although the event detection is parameterized, it might be difficult to fine-tune and adapt parameters to a wide range of events. As a result, we will focus our future efforts on

learning techniques that can be used to parameterize the system accurately for a wide range of criminal activities.

REFERENCE

[1] W. Zajdel, J.D. Krijnders, T. Andringa, and D.M. Gavrila, "CASSANDRA: audio-video sensor fusion for aggression detection", Proceedings of IEEE Conference on Advanced Video and Signal Based Surveillance, London, September, 2007, pp. 200-205.

[2] C.G.M. Snoek and M. Worring, "Multimodal Video Indexing: A Review of the State-of-the-art", Multimedia Tools and Applications, Vol. 25, pp.5-35, 2005.

[3] J. Kang, F. Lv, R. Nevatia, I. Cohen and G. Medioni, "Automatic Tracking and Labeling of Human Activities in a Video Sequence", Proceedings of the 6th IEEE International Workshop on Performance Evaluation of Tracking and Surveillance, Prague, Czech Republic, May, 2004.

[4] J. Kang, I. Cohen and G. Medioni, "Persistent Objects Tracking Across Multiple Non Overlapping Cameras", IEEE Workshop on Motion and Video Computing, MOTION'05. Breckenridge, Colorado, Jan 4-5, 2005.

[5] L. Yaroslavsky, B. Fishbain A. Shteinman and Sh. Gepshtein, "Processing and Fusion of Thermal and Video Sequences for Terrestrial Long Range Observation Systems", Proceedings the 7th International Conference on Information Fusion, pp. 848-855, Stockholm, Sweden, June, 2004.

[6] B. Fishbain, L.P. Yaroslavsky and I.A. Ideses, "Spatial, Temporal, and Inter-channel Image Data Fusion for Long-Distance Terrestrial Observation Systems", Advances in Optical Technologies, Vol. 2008, Article ID 546808, 2008.

[7] M. Valera, and S.A. Velastin, "Intelligent distributed surveillance systems: a review". IEEE Proceedings of Vision, Image and Signal Processing, vol. 152, pp.192-204, April, 2005.

[8] K. Toyama, R. Logan, and A. Roseway, "Geographic Location Tags on Digital Images", Proceedings of the ACM International Conference on Multimedia, pp.156-166, 2003.

[9] A. Pigeau and M. Gelgon, "Building and Tracking Hierarchical Geographical & Temporal Partitions for Image Collection Management on Mobile Devices", Proceedings of the ACM International Conference on Multimedia, 2005.

[10] I. Simon and S.M. Seitz. "Scene Segmentation Using the Wisdom of Crowds", Proceedings of European Conference on Computer Vision, Marseille, France, 2008.

[11] Y. Li, N. Snavely, D.P. Huttenlocher, "Location Recognition using Prioritized Feature Matching", Proceedings of European Conference on Computer Vision, Crete, Greece, 2010.

[12] S. Agarwal, N. Snavely, I. Simon, S.M. Seitz, and R. Szeliski, "Building Rome in a Day", Proceedings of International Conference on Computer Vision, 2009.

[13] T. Zhao, R. Nevatia, B. Wu, "Segmentation and Tracking of Multiple Humans in Crowded Environments", IEEE Transactions on Pattern Analysis and Machine Intelligence, 30(7):1198-1211, July, 2008.

[14] B. Wu and R. Nevatia, "Detection and Tracking of Multiple, Partially Occluded Humans by Bayesian Combination of Edgelet based Part Detectors", International Journal of Computer Vision, Vol. 75, pp.247-266, 2007.

[15] G. Medioni, I. Cohen, F. Bremond, S. Hongeng, R. Nevatia, "Event detection and analysis from video streams", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.23, pp.873-889, 2001.

[16] S.A. Ay, R. Zimmermann, and S. Kim. "Viewable Scene Modeling for Geospatial Video Search", Proceedings of the ACM International Conference on Multimedia, pp. 309-318, 2008.

[17] S.A. Ay, L. Zhang, S. Kim, M. He and R. Zimmermann, "GRVS: a georeferenced video search engine", ACM Multimedia 2009.

[18] S. Kim, S.A. Ay, R. Zimmermann, "Design and Implementation of Geo-Tagged Video Search Framework", Journal of Visual Communications and Image Representation Special Issue on Large Scale Image and Video Search; Challenges, Technologies, and Trends, 21(8):773-786, 2010.

[19] Y. Cai, T. B. Dinh, and G. Medioni, "Towards a practical ptz face detection and tracking system", In WACV, 2013.

[20] C. Huang and R. Nevatia, “High performance object detection by collaborative learning of joint ranking of granules features”, In CVPR, pages 41-48, 2010.
[21] B. Yang and R. Nevatia, “Online learned discriminative part-based appearance models for multi-human tracking“, In ECCV, 2012.
[22] C. Huang, B. Wu, R. Nevatia, “Robust Object Tracking by Hierarchical Association of Detection Responses”, In ECCV, 2008.
[23] Ali Khodaei, Cyrus Shahabi, and Chen Li, “Hybrid Indexing and Seamless Ranking of Spatial and Textual Features of Web Documents”, 21st International Conference on Database and Expert Systems Applications (DEXA10), Bilbao, Spain , August 2010.
[24] Ali Khodaei, Cyrus Shahabi, and Chen Li, “SKIF-P: a point-based indexing and ranking of web documents for spatial-keyword search”, Geoinformatica, Publisher: Springer Netherlands, Issn: 1384-6175, Doi: 10.1007/s10707-011-0142-7, October 2011.
[25] Ali Khodaei, Cyrus Shahabi, and Amir Khodaei, “Temporal-Textual Retrieval: Time and Keyword Search in Web Documents”, International Journal of Next-Generation Computing, November 2012.
[26] Chi Y., Wang H., Yu P. S., Muntz R. R. (2006): “Catch the Moment: Maintaining Closed Frequent Itemsets in a Data Stream Sliding Window”, Knowledge and Information Systems,10(3): 265-294.

Cyrus Shahabi

Cyrus Shahabi is a Professor of Computer Science and Electrical Engineering and the Director of the NSF's Integrated Media Systems Center at the University of Southern California. He was also the CTO and co-founder of a startup, Geosemble Technologies, which was acquired in June 2012. He authored two books and more than two hundred research papers in the areas of databases, GIS and multimedia. He served on the editorial board of IEEE TKDE and TPDS. Shahabi is an IEEE Fellow, and a recipient of the ACM Distinguished Scientist award and the U.S. Presidential Early Career Awards for Scientists and Engineers.

Seon Ho Kim

Dr. Seon Ho Kim is a computer scientist currently working in the Integrated Media Systems Center (IMSC) at the University of Southern California (USC). Before joining IMSC, he had worked at the University of Denver and the University of the District of Columbia as a faculty member for eleven years since he received his Ph.D. in Computer Science from the University of Southern California in 1999. He also received his BS degree in Electronic Engineering from the Yonsei University, Seoul, Korea in 1986, and M.S. in Electrical Engineering from the University of Southern California in 1994. Dr. Kim's primary research interests include multimedia servers, storage systems, databases, GIS, and mobile media applications.

Luciano Nocera

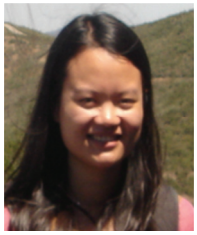
Dr. Nocera is a Computer Scientist and Associate Director of the IMSC at the USC and currently involved in research in geo-spatial decision making systems, video analytics, geospatial databases and serious games. Dr. Nocera has ten years of research project management experience andover fifteen years of programming experience with object oriented design. He has participated in research in computer graphics, computer vision, geospatial databases, immersive audio, visual analytics, virtual environments and serious games as the co-director of the IMSC's Serious Games Laboratory. Prior to joining IMSC Dr. Nocera worked at Eyematic Interfaces Inc. where he lead the development of some of the company's award winning products and conducted research in biometrics, 3D authoring and animation with applications to mobile devices communication interfaces andgames. Dr. Nocera obtained his PhD at INRIA / University of Paris VII, France and subsequently held a postdoctoral position at the GRASP Laboratory at the University of Pennsylvania (UPENN) where he conducted research in medical imaging, reverse engineering and teleimmersion.

Giorgos Constantinou

Mr. Giorgos Constantinou is a graduate student at the Department of Computer Science, USC. He is currently pursuing a M.Sc. degree after receiving the Fulbright scholarship. Before joining USC, he was studying at University of Cyprus where he received his B.Sc. degree in Computer Science in June 2012. His broad research interests span the areas of Data Management, Spatial Crowdsourcing and Mobile Computing.

Ying Lu

Ying Lu is a Ph.D. student at the Department of Computer Science, University of Southern California, under the supervision of Prof. Cyrus Shahabi. Before coming to the USC, she got her master degree from Renmin University of China, Beijing, 2012. Her research interests are spatial databases, location-based services, keyword search, and Mobile Computing.



Yinghao Cai

Dr Yinghao Cai is a research associate working in the IMSC at the University of Southern Califoria. Before joining IMSC, she worked in University of Oulu, Finland since she received her PhD in Computer Science from National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences in 2009. Dr Cai's primary research interests include computer vision, image/video processing and machine learning.

Gérard Medioni

Prof. Gérard Medioni received the Diplôme d'Ingenieur from ENST, Paris in 1977, a M.S. and Ph.D. from the University of Southern California in 1980 and 1983 respectively. He has been at USC since then, and is currently Professor of Computer Science and Electrical Engineering, co-director of the Institute for Robotics and Intelligent Systems (IRIS), and co-director of the USC Games Institute. He served as Chairman of the Computer Science Department from 2001 to 2007. Prof. Medioni lis an IEEE Fellow. He has made significant contributions to the field of computer vision. His research covers a broad spectrum of the field, such as edge detection, stereo and motion analysis, shape inference and description, and system integration.

Ramakant Nevatia

Prof. Nevatia received the PhD degree in electrical engineering from Stanford University, Palo Alto, California. He has been with the University of Southern California, Los Angeles, since 1975, where he is currently a professor of computer science and electrical engineering and the director of the Institute for Robotics and Intelligent Systems. Prof. Nevatia is an IEEE Fellow. His recent research interests cover a broad spectrum of object detection and tracking, and activity recognition.

Farnoush Banaei-Kashani

Dr. Banaei-Kashani is currently a research scientist at the Computer Science Department, University of Southern California (USC), where he also earned his PhD in Computer Science in 2007. Dr. Banaei-Kashani is passionate about performing fundamental research toward building practical, large-scale data-intensive systems, with particular interest in data-driven decision-making systems. He has published more than 40 referred papers and has received several awards. He frequently serves as a program committee member of highly ranked database conferences (including ICDE, VLDB and ACMGIS), and has also chaired the IWGS geo-stream data processing workshop (a subsidiary of ACMGIS) over last few years.