Fall 2011

Video event detection framework on large-scale video data

Dong-Jun Park
University of Iowa

VIDEO EVENT DETECTION FRAMEWORK ON LARGE-SCALE VIDEO DATA

by

Dong-Jun Park

An Abstract

Of a thesis submitted in partial fulfillment of the
requirements for the Doctor of Philosophy
degree in Computer Science
in the Graduate College of
The University of Iowa

December 2011

Thesis Supervisor: Associate Professor David Eichmann

ABSTRACT

Detection of events and actions in video entails substantial processing of very large, even open-ended, video streams. Video data present a unique challenge for the information retrieval community because properly representing video events is challenging. We propose a novel approach to analyze temporal aspects of video data. We consider video data as a sequence of images that forms a 3-dimensional spatiotemporal structure, and perform multiview orthographic projection to transform the video data into 2-dimensional representations. The projected views allow a unique way to represent video events and capture the temporal aspect of video data. We extract local salient points from 2D projection views and perform detection-via-similarity approach on a wide range of events against real-world surveillance data. We demonstrate that our example-based detection framework is competitive and robust. We also investigate synthetic example driven retrieval as a basis for query-by-example.

Abstract Approved: ________________________________
Thesis Supervisor

____
Title and Department

____
Date

VIDEO EVENT DETECTION FRAMEWORK ON LARGE-SCALE VIDEO DATA

by

Dong-Jun Park

A thesis submitted in partial fulfillment of the
requirements for the Doctor of Philosophy
degree in Computer Science
in the Graduate College of
The University of Iowa

December 2011

Thesis Supervisor: Associate Professor David Eichmann

Graduate College
The University of Iowa
Iowa City, Iowa

CERTIFICATE OF APPROVAL

PH.D. THESIS

This is to certify that the Ph.D. thesis of

Dong-Jun Park

has been approved by the Examining Committee for the thesis requirement for the Doctor of Philosophy degree in Computer Science at the December 2011 graduation.

Thesis committee:

David Eichmann, Thesis Supervisor

Joseph Kearney

Nick Street

Juan Pablo Hourcade

Michael Mackey

ACKNOWLEDGEMENTS

It is a pleasure to thank the many people who made this thesis possible.

Foremost, I would like to express my sincere gratitude to my advisor Prof. David Eichmann for the continuous support of my Ph.D study and research, for his patience, motivation, enthusiasm, and immense knowledge. His guidance helped me in all the time of research and writing of this thesis.

Besides my advisor, I would like to thank the rest of my thesis committee: Prof. Joseph Kearney, Prof. William Street, Prof. Juan Pablo Hourcade, and Prof. Michael Mackey for their encouragement, insightful comments, and hard questions.

I owe my loving thanks to my wife Jaehee Kim and my son Ian Park. They have lost a lot due to my research abroad. Without their encouragement and understanding it would have been impossible for me to finish this work. Lastly, and most importantly, I wish to thank my parents, Okja Kim and Youngmyung Park. They bore me, raised me, supported me, taught me, and loved me. To them I dedicate this thesis.

ABSTRACT

Detection of events and actions in video entails substantial processing of very large, even open-ended, video streams. Video data present a unique challenge for the information retrieval community because properly representing video events is challenging. We propose a novel approach to analyze temporal aspects of video data. We consider video data as a sequence of images that forms a 3-dimensional spatiotemporal structure, and perform multiview orthographic projection to transform the video data into 2-dimensional representations. The projected views allow a unique way to represent video events and capture the temporal aspect of video data. We extract local salient points from 2D projection views and perform detection-via-similarity approach on a wide range of events against real-world surveillance data. We demonstrate that our example-based detection framework is competitive and robust. We also investigate synthetic example driven retrieval as a basis for query-by-example.

TABLE OF CONTENTS

CHAPTER

LIST OF TABLES

Table

LIST OF FIGURES

Figure

CHAPTER 1
INTRODUCTION TO VIDEO EVENT DETECTION

1.1 Motivation

Advancement in technology for digital acquisition of video has led to an increase in needs for automatic event detection. Large collections of digital videos are now commonly seen in many areas such as commerce, government, academia and surveillance. This abundance of video data genuinely leads to user requirements for accessing temporal points of interest that matches user needs. Applications such as content-based retrieval, video summarization and surveillance for security drive these types of requirements. However, searching such collections based on visual contents is enormously challenging.

Current classification and retrieval techniques for video data cannot be directly translated and applied to large-scale video collections. First, in today's society, multimedia data is everywhere and generated rapidly. With the explosion of video data, the task of video retrieval is becoming increasingly difficult [48]. The size of video data tends to be huge, and the rapid generation of video data creates special challenges for storage, annotation and retrieval. Second, accessing the content of video data is an inherently time consuming process due to the streaming nature of video data. A video event can only be understood through a sequence of images, and this multi-dimensionality makes video retrieval hard and time-consuming. This *temporal aspect* has not been adequately addressed in most retrieval systems [98], which prohibits a true understanding of video events. Third, the user's information need is dynamic, and the restricted approach with specialized parametric models may not be desirable for certain real-world applications [137].

Consider the case of the British Broadcasting Corporation (BBC) seeking capability to automatically index and retrieve their archive of programs [32]. They currently have 750,000 hours of video data in their archive, producing an 700 additional hours of programming each week. They are handling about 2000 enquiries each week to locate certain video segments that satisfies the user's information needs. Now imagine an automated system that can search for video events with such a dynamic data. This poses the challenging tasks not only for storage but also for video annotation, indexing and retrieval. Having huge archives of video collection is hardly of any benefit if we have no effective means of locating video clips which are of relevance to our information needs. Due to the sheer volume and complexity of data, most video retrieval systems treat video as collections of still images and extract relevant low-level features from selected keyframes to compare visual contents [98]. However, video events can only be understood by examining temporal characteristics. Therefore, video retrieval systems need to address the following criteria to perform adequate event-based retrieval.

- Scalability: Is the approach applicable to large-scale data?
- Competency: Does the approach include temporal aspects of video data for event retrieval?
- Robustness: Is the approach able to retrieve a wide range of video events?

Based on these three research issues, we conjecture that one of the main challenges for video event retrieval is properly *representing video events*. How can we represent the visual changes over time that are induced by a certain motion, which allows fast scanning of existing data as well as a high degree of adaptability to new video events? The instance of human interaction or motion trace needs to be effectively captured for detection in later stages. The current state of video retrieval research is not fully accommodating these unique multi-dimensional data characteristics [98].

This thesis investigates a novel framework for representing and detecting temporal video events that are useful in video applications on large-scale collections. Most existing vision systems approach this problem by modelling each individual motion for classification. This is hardly applicable in large-scale video data. On the other hand, many video retrieval systems rely on a few highly undescriptive sets of features such as MPEG motion vectors for retrieval purposes. The objective of this thesis is to develop temporal video event representation methodologies that are flexible enough to generalize to a wide range of video events on large-scale video data. This work was motivated by the traditional content-based image retrieval studies where visual cues from a 2D image plane are used as a descriptor to analyze 3D objects in real-world space.

1.2 Problem statement

We consider video data as a streaming series of images without any temporal boundaries such as shots or scenes. Such structural information is often used for temporal windows in detection systems. Temporal boundary annotations may be available, but our system does not make such assumptions even though it can be readily applied on such cases. A video event denotes an observable action in a video stream. In this thesis, we are only considering temporal video events where there is an observable change of state. The goal of this thesis is to develop a succesful methodology for tasks with a temporal rather than a spatial extent. The scope of the thesis is limited to fixed camera positions, leaving the more advanced case of a moving camera to future work. Motivated by the TRECVid 2008 event detection task, we take their definition of events and method of evaluation, and evaluate our performance against real-world surveillance data.

1.3 Thesis outline

Chapter 2 discusses the literature related to the two main classes of research in this thesis: event classification in the computer vision literature and event detection in the video retrieval literature. We also briefly discuss the recent development of the TRECVid event detection task. Chapter 3 presents the overview of our framework. The spatiotemporal projection methodology is described in Chapter 4. Chapter 5 explains the framework for event detection and presents an evaluation performed on real-world video events. Chapter 6 presents a retrieval task performed with synthetic examples. Finally, Chapter 7 provides a summary and suggests possible future research directions.

CHAPTER 2
RELATED LITERATURE

2.1 Introduction

Video is a rich source of information, and research topics on video information science encompass a wide range of research disciplines. One generally starts from issues regarding data acquisition, automatic structuring, management, annotation, browsing, and user interface. In the issue of understanding events within temporally varying visual data, the general research approaches can be divided into two separate yet enriching disciplines. Computer vision generally models individual motion for classification by analyzing a sequence of static information. The dataset naturally tends to be more controlled to ensure said motion is visible in a constrained manner. This commonly results in significantly overstating performance unless analyzing open domain sources. Video retrieval traditionally adopts extensively lossy sets of features extracted from a sequences of images and compares them to similarly retrieved video clips. Computational time is an important factor, and less-constrained data sets invite separate challenging tasks such as annotation and automatic structuring.

Visual information differs from traditional text information in that the raw format of visual data does not provide any inherent semantic meaning. The digitized visual data consists purely of arrays of pixel intensities, and this information can not be readily interpreted into a proper semantic concept. This semantic gap is the main difficulty of developing visual information systems [116], and a major goal of visual information research is to reduce this gap [100]. One of the key issues is to extract useful features from the raw data that can help understand the visual data's semantic contents. When the description of content is extracted from a single image plane, the resulting spatial representation includes colors [122, 84, 121], edges [11, 109] and textures [45, 123, 49]. Recent trends include capturing local salient points by processing the local geometrical properties [74]. It is important to note that the goal of the description of content is *not* to describe its content in its entirety. Rather, its goal is to provide means to compare to extract *similar* data [116].

2.2 Motion modelling in computer vision

Human motion understanding from a sequence of images has been studied actively in computer vision. This is a very popular research area that involves hundreds of publications. For more in depth review, there are several survey papers available such as [1, 80, 125]. For instance, Moeslund *et al.* highlight more than 300 research papers from 2000 to 2006 in vision-based motion capture and analysis [80]. Computer vision generally denotes any research activities enabling the machine to extract some information from visual data to perform vision-related tasks. In this thesis, we limit ourselves to vision tasks in human motion analysis or event detection with a temporal extent. Motion is analyzed from a sequence of images to produce information based on the apparent motion in the images. The appropriate set of features such as optical flow are extracted, and the temporal extent of motion is captured using a differential approach [52, 51] or background segmentation [79]. The algorithms are rarely tested in real-world large-scale data which tends to be less constrained. The controlled environment tends to overstate performance, and it is common to see reliably high detection ratios in this domain.

Closely related visual appearance-based research areas include the motion-based approach [10, 28, 103] and the spatiotemporal volumetric approach [31, 135, 9, 136]. The motion-based approach detects events based on motion-induced patterns on image frame plane. The spatiotemporal approach captures motion specific 3D pattern within 3D spatiotemporal volume generated by stacking a sequence of images.

In Bobick and Davis' work [10], a temporal template is constructed using a motion-energy image (MEI) and a motion-history image (MHI). An MEI is a binary cumulative motion image that shows a range of motion within a sequence of images. An MHI shows the temporal history of motion where more recently moving pixels are brighter. Thus, this MHI implicitly represents the direction of movements. Sadek *et al.* combined the differential approach with the template approach [103]. They computed the frame-to-frame difference and extracted shape moment descriptors along with the temporal motion trajectory. They used SVM classifiers to detect motion events. In [28], Dong *et al.* proposed a pointwise motion image for event representation by performing foreground segmentation and establishing pointwise correspondences between frames. In [104], Saligrama *et al.* presented a statistical model to detect abnormal behavior using Markov chains. Motion images from multiple cameras were aggregated to produce anomaly image. In another anomaly detection work in [53], Jiang *et al.* tracked all objects in the spatiotemporal volume and proposed data mining approach to detect abnormal events by modelling each object's appearance. Often, the approaches are not suitable for application to video retrieval on large-scale corpora since they rely on well-constructed video segments (e.g., the segment showing a single motion).

Spatiotemporal approaches take a streaming video as spatiotemporal 3D volume by stacking a sequence of images. The motion performed within this 3D volume is treated as a 3D object in the spatiotemporal space [135, 9, 136, 127]. This framework requires a reliable object segmentation to form a spatiotemporal event object. Spatio-temporal interest points have been applied to detect significant local variations in both space and time to detect events [64, 27, 85, 96]. Similar to [85], Gong *et al.* applied the bag-of-words representation onto 3D spatiotemporal volume [37]. The spatio-temporal volume can be sliced to reveal patterns induced by moving objects [87, 83, 99]. Shechtman *et al.* used space-time correlation of the video with an action template [108]. To overcome the local segmentation issues, Ke *et al.* applied spatio-temporal volume features (such as optical flow) to scan video sequences in space and time [59]. Yan and Luo extracted a volume descriptor around a 3D spatiotemporal interest point [131]. In another work by Ke *et al.*, mean shift segmentation was applied on spatiotemporal volumes, where over-segmented regions indicated spatiotemporal volumes with motion information [60]. Zelnik-Manor and Irani recognized the need for simpler behavioral distance measure to capture events with different spatio-temporal extents and applied unsupervised event clustering based on behaviors [137].

2.3 Video retrieval

Video information retrieval (VIR) typically deals with unconstrained video data obtained from real-world scenarios. These video data are loosely structured without temporal boundaries, i.e., no pages, paragraphs or chapters. A video exhibits a streaming sequence of images without a notion of hierarchical structure. The early stages of video retrieval research focused on the task of partitioning the video into physical meaningful units. A shot in the video is the basic component of a video stream and is defined to be a sequence of consecutive frames taken contiguously by a single camera [44]. Once the video is segmented into shots, a keyframe is typically extracted for each shot and indexed into a database. This can provide compact abstraction for video indexing, browsing and retrieval [69]. However, it is not always the case that video data can be segmented in some meaningful way. For instance, surveillance video is recorded without going through camera changes. In this case, a predetermined number of frames can be defined as a shot (i.e., 100 frames), or a single frame can be an indexing unit.

One of the main future directions of video information retrieval (VIR) is the collaboration between end users, academic researchers and private industry to promote the growth of the multimedia search field [67]. The resulting LSCOM (Large-Scale Concept Ontology for Multimedia) is a joint effort between users, knowledge experts and researchers to standardize what set of semantic concepts the research community should focus on [82]. While this definition cannot possibly address every single semantic concept users will need in the future, it shows a sufficient variety of concepts that system designers can expect users to want their systems to represent. Currently, the LSCOM effort includes an ontology of almost 1000 semantic concepts. Each concept falls into one of a small number of broader categories, including events, objects, people and program. There is also a light scale LSCOM version that was used for TRECVid that includes 7 dimensions and 44 concepts. Table 2.1 shows the list of the light scale LSCOM that was used for TRECVid 2005.

Table 2.1: Examples of light scale LSCOM definition

Dimension	Concepts
Program	Politics, Business, Science, Sports, Weather
Setting	Indoor, Court, Office, Meeting, Outdoor
People	Crowd, Face, Person, Police, Military
Objects	Animal, Airplane, Car, Bus, Truck
Activities	Running, Marching
Events	Explosion, Natural Disaster
Graphics	Maps, Charts

One interesting point is the distinction between keyframe-based and video-based concepts [82]. Due to computationally expensive video processing, most video retrieval systems treat video as a collection of still images and extract relevant low-level features from selected keyframes to compare visual contents [41]. Even though this single keyframe is sufficient for some static concepts, event concepts require the understanding of temporal characteristics. Spatiotemporal information in video captures the gradual transition of the spatial object that changes over time. From the LSCOM definition, concepts such as airplane crash, airplane takeoff, dancing

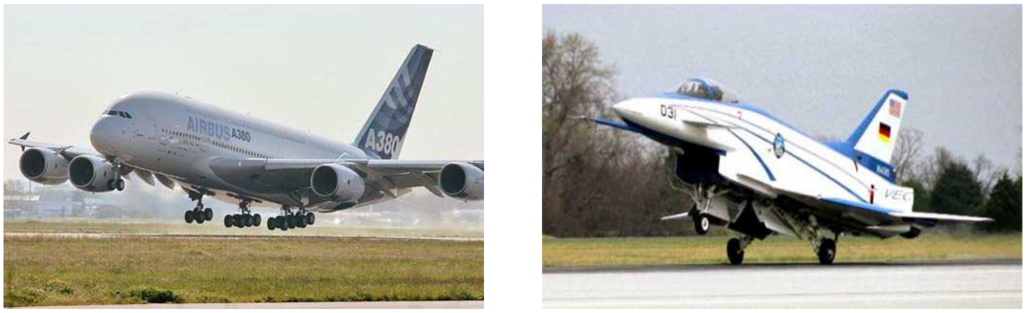


Figure 2.1: Take Off or Landing?

or car exiting require spatiotemporal modeling . In Figure 2.1, the importance of temporal aspects is clearly shown. Even though both images provide enough visual information to discern what is present inside those images, an airplane, the lack of temporal information prohibits a true understanding of visual data (i.e., taking off or landing?). The introduction of event concepts as a part of the LSCOM definition clearly shows the need for features that can enhance event-based retrieval in the temporal domain.

In the context of video information retrieval, temporal modeling by analyzing the sequence of images is a relatively new research area [98]. It is not yet straightforward as to how spatiotemporal information can be represented. After reviewing almost 300 publications on the subject of video retrieval, Snoek and Worring state that the use of temporal feature modelling is not common in content-based video retrieval due to their computational costs [118]. Ignoring the temporal aspect of video indicate that we are still doing content-based image retrieval (CBIR) and not true video retrieval [115]. For instance, one of the longest standing and active video research programs, Informedia, does not list a temporal feature in their feature list [46]. This is partly due to the fact that CBIR is in much more mature stage, compared to video retrieval, and the CBIR techniques can be directly applied on this domain. This leads the researchers to generally focus on the concept categorization and spatial semantics [119, 129]. It has been observed that the temporal features of video data was historically ignored in TRECVid, leading to low performance in detecting events and actions in the data (D. Eichmann, Personal Conversation, 2005). Recently, the importance of events as a semantic concept has been acknowledged, and TRECVid has begun to include these type of concepts as a part of their workshop sessions [2].

Clearly, an event requires more than a single frame to fully understand its semantic meaning, which implies the analysis of features extracted from multiple frames. Often, temporal modelling is achieved by highly primitive sets of features from short video segments. Ardizzone and La Cascia extracted a gradient-based optical flow field from a keyframe (and a few frames before and after this keyframe) in defining a motion-based descriptor [4]. Chang *et al.* created VideoQ as a video search engine supporting spatiotemporal queries [18]. This system captures motion information by segmenting a moving object and constructing its motion vector. However, for content-based retrieval, automatic segmentation is not used often in broad domains because the technique is computationally expensive and brittle [116]. The IBM system employs motion vectors from P- and B-frames of the MPEG-encoding protocol [14]. In [91], Pers *et al.* constructed a histogram usng optical flow to describe the dominant motion from a video sequence. In [5], Bakheet *et al.* applied neural network on logpolar histogram using spatially captured interest points.

The tracking-based approach aims to model motion by capturing the location of a moving object over time on the image plane. Sivic *et al.* applied tracking of local salient points to group video shots based on object appearance [113]. They built an implicit representation of the 3D structure of objects by merging tracking results from different views of the same object. Even though they attempted to model the temporal aspect by tracking, their application was limited to object-level categorization. In [93], Pogalin *el al.* derived general measures of the activity based on the notion of periodicity in the scene. They aligned the object windows by using a tracking algorithm and detected events based on spatially coherent changes over time. Jung *et al.* applied clustering onto tracking results based on global motion information [55]. In each cluster, they built 4D histogram by counting the individual object's location and velocity (i.e., x and y direction for location and velocity). In [63], Lai *et al.* extracted interest points in consecutive frames and tracked the matching points between frames to detect motion. Kaaniche and Bremond applied another local descriptor, Histograms of Oriented Gradient, to extract spatial description and tracked over multiple frames to detect gestures [56]. Chen *et al.* performed human tracking and applied hidden Markov models to detect events in TRECVid surveillance data [19]. Their performance indicates that our approach is competitive and robust.

Several researchers have considered the trajectory of an object as a 2D curve on a plane [13]. Little and Gu used the path and speed trajectories to record an object's motion [71]. The trajectory curve matching is performed on the maximum curvature feature points, angles between successive segments and the relative lengths of adjacent segments. Dagtas *et al.* use a trail-based model, capturing the motion of salient objects over a sequence of frames [21]. The binary trajectory image is used in absolute search, spatial-invariant search and scale-invariant search. Chen *et al.* proposed a distance function for trajectory matching, called Edit Distance on Real Sequences (EDR) [20]. This is based on the Edit distance between two strings, which is the number of operations required to transform one set into the other. They argued that their approach is robust to noise and suited better to the local trajectory shift. In [47], Hsieh *el al.* combined the trajectory curve matching and the string matching approach. In [75], Ma *et al.* described the motion trajectory as a set of lowband frequency coefficients. The issue with trajectory matching is that once the motion is reduced to motion path on xy plane, the motion detail is lost. Some events that require more information than a simple trajectory will need a detailed representation in temporal space.

Similar to the spatiotemporal approach found in computer vision, some researchers have applied spatiotemporal methods on to retrieval framework. Dyana and Das [30] proposed a spatiotemporal representation called MST-CSS (Multi-Spectro-Temporal Curvature Scale Space). The spatial and temporal feature combination was achieved by a series of filtering processes on the spatiotemporal volume. The spatiotemporal volume is built by foreground segmentation of the moving object, and its shape is extracted from the median frame in the volume. The motion trajectory is extracted by tracking the centroid of the foreground object. We can see that the approach is quite similar to what we have seen in computer vision approach. It is not clear how this can be applied in large-scale data with very high number of individual objects (and their interactions). Gao and Yang extracted the interest points from 2D spatial objects and tracked them in 3D spatiotemporal volume [36]. However, their work was limited to 2D object detection. Jin and Shao applied bag-of-words (BOW) representation with local interest points to event detection [54]. Interest points was extracted from 3D spatiotemporal volume, and the motion is described by BOW histograms. However, much of their work was evaluated with highly controlled set of dataset, and it remains unclear how it will perform in large-scale data.

Some researchers have focused on the semantic modelling of video data. Dao and Babaguchi [23] applied Allen's Interval Algebra for temporal description of events. Their work was aimed at semantic modelling of various types of events by capturing visual and text features. Singh *et al.* proposed another semantic level modelling approach to spatiotemporal video analysis [112]. They extracted video object information by segmentation and tracking, and the video shot is described by a string to capture semantic relationships. Individual motion is not the focus of these approaches. They generally study the feature-level reduction of the video shot to measure shot-to-shot comparison.

2.4 TRECVid-based event detection

Automatic event detection in the context of video retrieval on large-scale data can be found in the recent TRECVid workshops [2]. They began to offer an event detection task in 2008 as a part of a pilot program. For this task, the corpus consists of 100 hours of surveillance video data obtained at the London Gatwick airport: 10 collection sessions x 2 hours per session x 5 camera locations. The data exhibits a constant background (e.g., fixed camera location with no camera movement). The corpus was divided into 2 parts for training and testing purposes. Many participants noted that noisy video with relatively high traffic makes tracking quite difficult [120, 134, 57, 133, 66, 130]. Also, slight changes in light condition pose an interesting problem for the participants because it is not clear what the light condition will be for the testing set.

The event set included *PersonRuns, CellToEar, ObjectPut, PeopleMeet, PeopleSplitUp, Embrace, Pointing, ElevatorNoEntry, OpposingFlow, and TakePicture.* The corpus was annotated with event type, event starting time and event duration. Some events occur more frequently within the given corpus and are relatively easy to find while others pose a serious challenge even for human annotators. In total, 16 participants submitted completed runs for this task. The most popular events included *ElevatorNoEntry, OpposingFlow*, and *PersonRuns* while the least popular events were *CellToEar, Embrace, ObjectPut and Pointing.*

The participating groups exhibited very similar approaches to the problem, including optical flow, motion vector, background/foreground segmentation, and person detection/tracking. For example, one of the most active video retrieval research groups, Fudan University, argued that optical flow is too noisy for this corpus since the data exhibits a very complex environment [130]. Thus, they extracted a motion vector of a subblock, which is defined to be a sum of absolute value of pixel differences in a block between frames. To counteract the noisy data, they performed a series of erosions and dilations for given a video frame. They also tried to calculate a distance between camera and subblock to model the location of spatial objects.

The challenging set of events applied on large-scale realworld surveillance video data presents interesting research task. The evaluation results in Chapter 5 show that the task defined in this thesis is highly challenging and calls for much needed improvements.

2.5 Discussion

We claim here that modern state-of-the-art video retrieval systems need to answer the demanding criteria of scalability, competency and robustness in today's multimedia environments. The system needs to effectively handle large-scale data to retrieve a variety of video events. Most current systems focus on a feature-based approach that requires specialized parameter optimization. Automatic event detection is often reduced to a smaller niche problem where high precision and recall can be achieved. It is not straightforward to establish how a given feature relates to specific video events, and building a detector for new video events usually means a complex process for newly selected visual features and metrics. Dynamic information needs with large-scale data require a robust and scalable framework for video event retrieval. The spatiotemporal volumetric approaches discussed above model a video segment as a 3D shape to compute motion similarities. We propose alternatively to analyze video events as inherently lossy projections of 3D structure onto orthogonal 2D representations. We argue that this reduced form of data supports a robust and scalable approach that can be applied to large-scale data.

CHAPTER 3
OVERVIEW OF VIDEO DETECTION FRAMEWORK

Figure 3.1 shows a typical algorithm flow found in motion analysis work in computer vision systems. This also closely resembles most systems we have seen in TRECVid event detection task. The input video is a set of clips containing certain motions, which are divided into a number of classes. The system extracts features from video data and models each video clip as a collection of features. Each class of motion is processed in a learning phase, and a model is generated that best represents the distribution of each category of motion. In detection, a feature set is extracted from an unknown video clip, and the decision process identifies the category model that fits best the distribution of unknown video clip.

The common issue we first see is the general premise of these approches, which is substantially based on the notion of a classification framework. The computer vision approaches and also much of TRECVid event detection assume that the event examples are available from a set of precisely annotated clips. The model building process relies on the positive and negative examples given from training set. However, the large-scale evaluation efforts such as TRECVid show that annotation is both incomplete and inaccurate. This is partly because the annotation process itself is expensive, costing $10-15\times$ actual video time. The human factor coupled with the ambiguity of event guidelines results in incomplete and inaccurate annotation. This observation is backed by analysis done by TRECVid and Linguistic Data Consortium (LDC) [101]. They found that the miss probability of human annotators is very

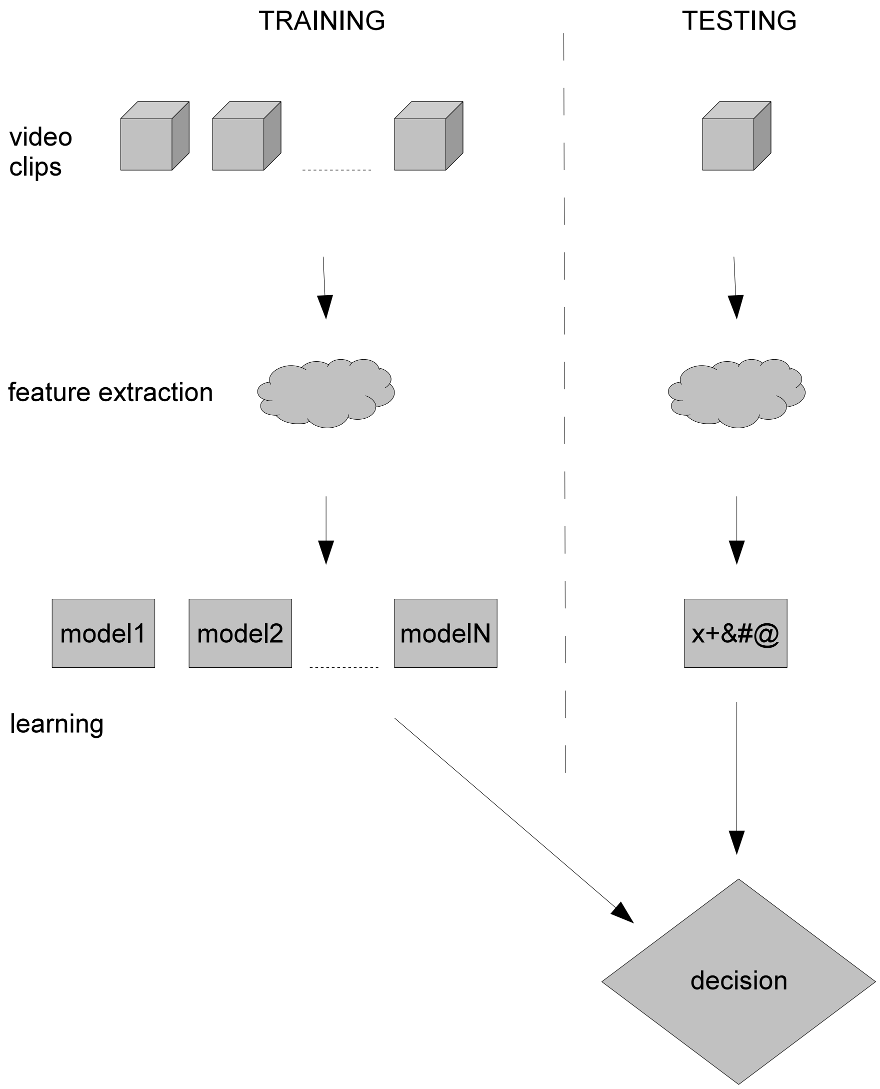


Figure 3.1: Flow chart of typical motion analysis algorithms

high (i.e. for 35 minutes of data, a single person has about 50% chance of a missed annotation when annotating five events). This brings up the question of the experimental setup of the classification approach on this type of data. When negative data are input into the learning phase, we cannot be sure that they are in fact negative examples. The annotation issue above implies that the negative examples could well be missed postive examples.

Additionally, video events can be considered independent but are not exclusive from each other. For instance, in real-world video data, the same video segment can contain multiple events - cell to ear, walking, getting into elevator, opposing flow, etc, - all at the same time. The individual motion modelling may be applied to the task where there exists precise and complete annotation both in the spatial and temporal extents. However, the current level of experimental setup only supports incomplete temporal annotation without any spatial description.

Much can be learned from traditional text retrieval systems such as internet search engines [89, 12]. In this domain, the system collects web pages and analyzes each page to determine how it should be indexed. The extent of analysis and indexing differs from one search engine to another. The general retrieval process finds relevant documents to user queries based on its set of measures obtained from natural language processing techniques such as tf-idf [105]. Statistical measures provide a basis for comparison between documents for indexing. Retrieval is based on similarity between a user query and a web page. We choose to adopt a similar approach and address the problem of video event detection on large-scale data as detection via similarity.

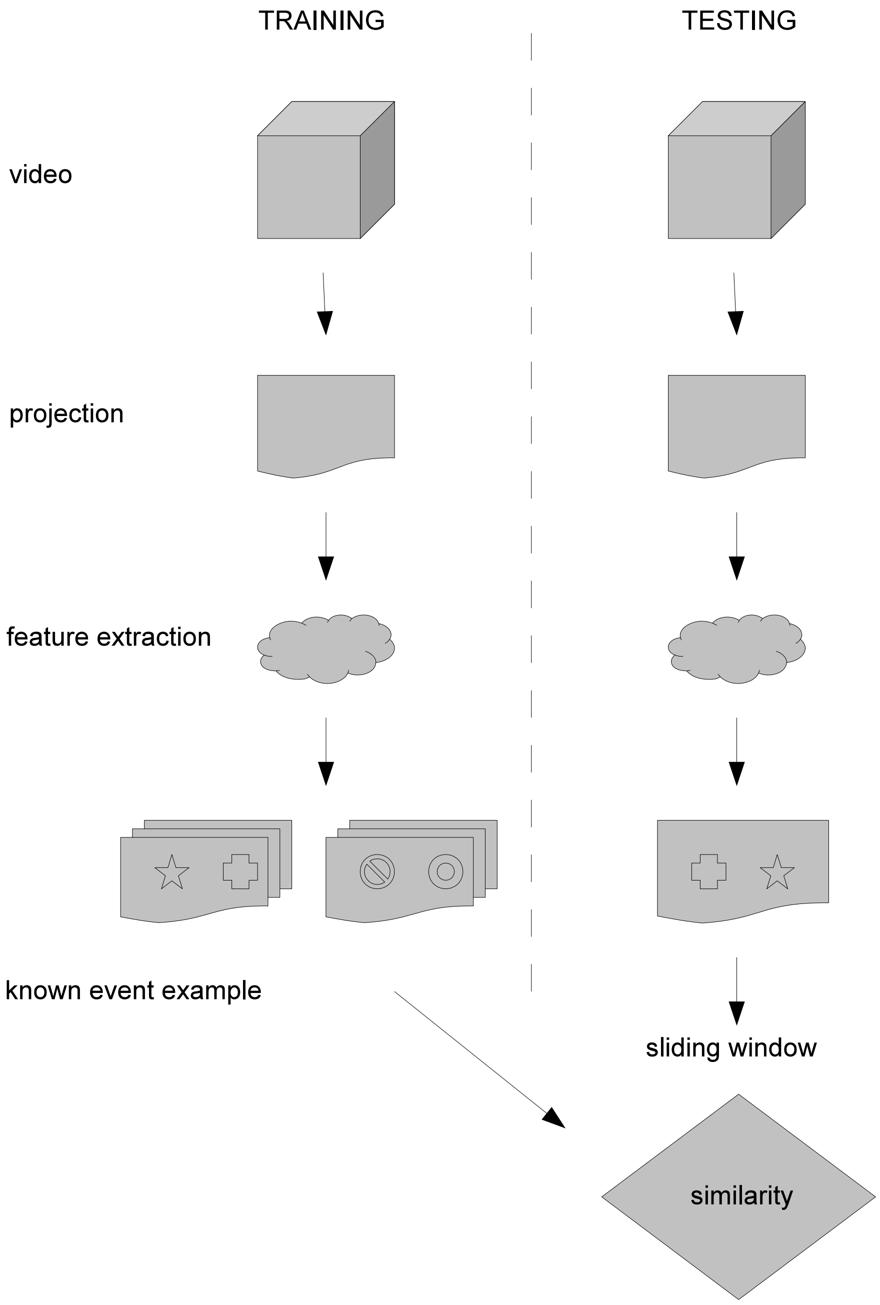


Figure 3.2: Flow chart of the algorithm

Figure 3.2 is a summary of our algorithm for a detection task. We first process input video data through a projection module, reducing the size of data and capturing the temporal characteristics. The system extracts features from the resulting 2D projection data. Each event has a collection of positive examples, summarized by a set of features. The unknown video stream also goes through projection and feature extraction. Since the video data has no temporal boundaries, we take a sliding window approach to capture similarity between known examples and unknown data. We acknowledge that our sequential approach is far from optimal. However, considering the limitations presented by semantic gaps and task complexity as discussed in Chapter 2, we demonstrate that our example-based approach is a step closer to fulfilling the requirements of the video event retrieval system.

CHAPTER 4
PROJECTION OF SPATIOTEMPORAL VOLUME

4.1 Introduction

Video data is a sequence of images representing 3-dimensional scenes in motion. The 3-dimensional volume is projected onto a 2-dimensional screen by video recording devices. As we can see from Figure 4.1, light rays from an object in 3D world are collected by a set of lenses and rendered onto a film surface. The camera transformation of the 3D world onto a 2D surface can be described as perspective projection. Light rays from an object pass through a lens and act as projectors. The surface upon which an image is formed is a projection plane. This type of projection exhibits the following characteristics: convergence of parallel lines, diminution of size and nonuniform foreshortening [16]. Photographic projection is performed repeatedly over time, and a stream of 2D images records the changes in scene. The projection function in this case exhibits binary characteristics. Since the light ray is projected onto a projection plane, light rays from occluded objects are blocked by foreground objects and not rendered. A 2D video of a 3D world inherently loses some information in the projection process. The complexity reduces significantly over the process while providing a nice summary of the 3D scene in motion.

4.1.1 Projection of 3D video volume

We propose here to represent video events using projection onto a spatiotemporal video volume, since the complexity of the video data prohibits effective event-

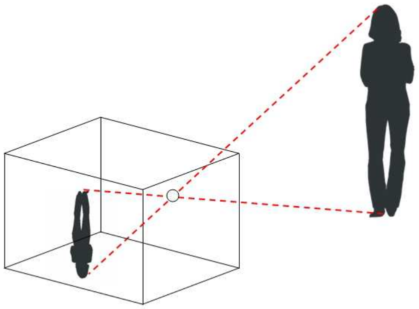


Figure 4.1: A camera capturing a 3-dimensional world onto a 2-dimensional surface can be illustrated by principle of a pinhole camera. Light rays converge through a lens and projected on a film surface. Modern camera systems still follow this early pinhole camera model.

based retrieval. There are various ways of transforming 3D objects onto a 2D surface. When it is performed with a straight line (projector) onto a plane, it is called planar geometric projection [16]. The projectors emanating from a single point called the center of projection intersect with a plane of projection. The kind of mapping protocol used decides the type of projection. As discussed in the previous section, video projection follows the perspective projection protocol. This type of projection allows the most realistic rendering of an object as seen by human eyes.

We take the video as a stream of images and view this as a 3D volumetric structure by stacking images. The objects in motion within the video stack can be viewed as a 3D structure. Volumetric shape description can be achieved by mapping into multiple 2D projections. The choice of the projection protocol depends on number of factors. Orthographic projection reveals the actual measurements of a 3D object and

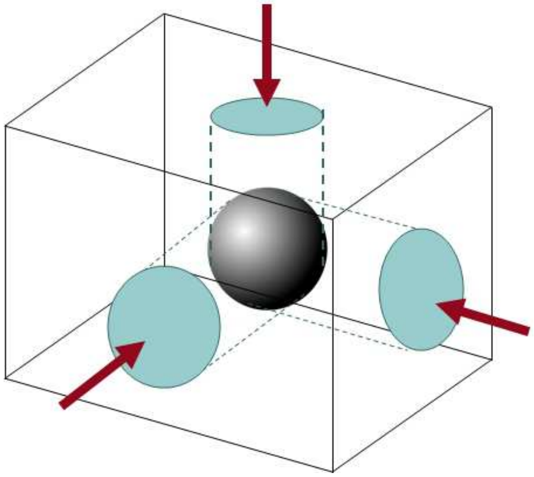


Figure 4.2: The example of multiview orthographic projection for a sphere. A 3D shape inside the stack can be projected onto 2D plane.

is often used in architectural drawings. This can be performed by surrounding the object with projection planes forming a rectangular box. Since a single orthographic projection involves only one perspective of an object, multiple projections are often needed. The number of projections depends on the complexity of an object. The most common choice is top, front and right side view [16]. An example is shown in Figure 4.2 for an orthographic projection. In this case, the simple 3D shape (i.e., a sphere) can be reconstructed perfectly from the mapped 2D shapes by analyzing the back-projections of 2D shapes from multiple camera views.

4.2 Radon projection

In this section, we propose a novel approach to incorporate the temporal aspect of video data using spatiotemporal volume. By projecting a 3D video volume onto

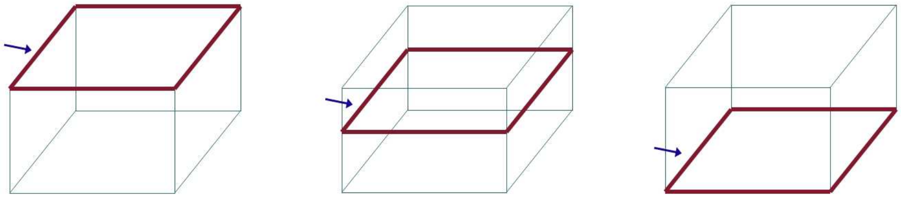


Figure 4.3: Frames at time $t = 0$, n and T where $0 \leq n \leq T$

2D, a video event is represented in concise form, which allows more effective retrieval in large scale video data.

In the following sections, we introduce the mathematical foundation of the project and the basic concepts of video spatiotemporal volume. For this approach, we assume that the video data is a stream of individual images without any other encoding information. *MPEG* video compression utilizes a series of operations that provide the *MPEG* motion vectors between frames. However, our approach takes the video data as a sequence of images, making this approach independent of the underlying data format.

4.2.1 Spatiotemporal volume

A single image is viewed as a 2D structure with width and height. A video stack is constructed by stacking a series of images, forming a 3D spatiotemporal structure. Figure 4.3 shows the 3D volumetric structure of the stack. The first image is at the top of the stack, and we put any subsequent images underneath previous images. In this thesis, we use the convention of 3D Cartesian coordinate notation, using the width of frame as x axis, the height as y axis and the time as t axis.

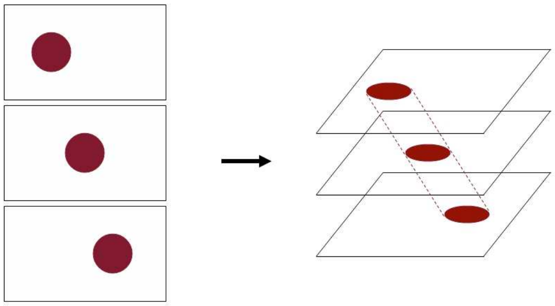


Figure 4.4: This three-frame sequence of a circle moving to right can be viewed as a three-dimensional cylindrical object within the stack. The gap between frames is exaggerated for illustration.

4.2.2 Orthographic projection of video stack

Suppose that physical objects in 3D space and their corresponding video stacks follow the following assumptions.

- The motions of physical objects are observed from a fixed camera. The background remains constant.
- The location, speed and direction of a physical object changes smoothly over time.
- The lighting does not change over time. The illumination of the object remains constant.

This set of assumptions allows the moving object to maintain relatively constant color and brightness over time. We regard the motion of the spatial object over time within spatiotemporal volume as 3D shapes induced by the contours in the spatiotemporal volume. Figure 4.4 shows a sequence of 3 frames depicting a circle

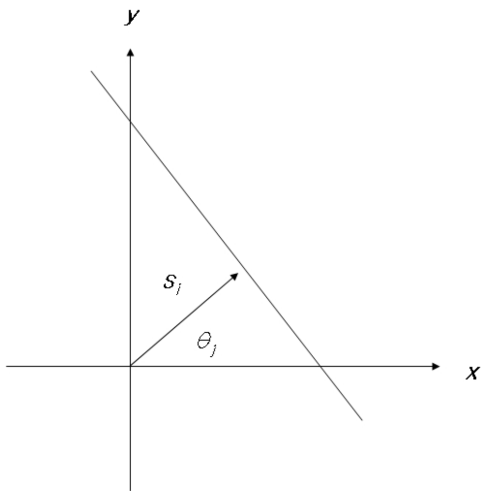


Figure 4.5: A straight line in normal representation

moving to right. In this case, the apparent displacement of a circle to right forms an inclined cylinder in the spatiotemporal volume, where the angle of inclination is determined by the rate of motion.

A 3D shape can be represented in 2D using multiview orthographic projection (MOP)[16]. The Orthographic Radon transform provides a very general tool to perform projection onto a signal of higher dimension [38]. In this case, the projection is dependent on the density of the volume that is being projected. The equation of a straight line with a distance s_i and an angular orientation θ_j relative to the origin of the coordinate system is:

$$x \cdot cos\theta_j + y \cdot sin\theta_j = s_i. \tag{4.1}$$

This represents a single projection beam onto a volume. The Radon transform specifies the accumulator functions for the projection ray. By using the sifting property of the Dirac impulse δ, the ray sum given by this line can be expressed by an integration

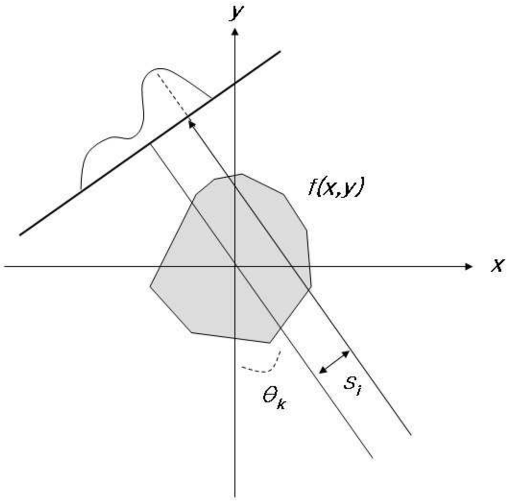


Figure 4.6: The Radon Transform

along the line,

$$g\left(\theta_j, s_i\right) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f\left(x,y\right)\delta\left(xcos\theta_j + ysin\theta_j - s_i\right)dxdy. \tag{4.2}$$

If we consider all values of s and θ, this defines the continuous 2D Radon transform of $f\left(x,y\right)$:

$$g\left(\theta, s\right) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f\left(x,y\right)\delta\left(xcos\theta + ysin\theta - s\right)dxdy. \tag{4.3}$$

In the discrete case with $W \times H$ images, Equation 4.3 becomes:

$$g\left(\theta, s\right) = \sum_{x=0}^{W-1}\sum_{y=0}^{H-1} f\left(x,y\right)\delta\left(xcos\theta + ysin\theta - s\right). \tag{4.4}$$

Thus, we can see that projection accumulates the pixels of $f\left(x,y\right)$ along the line defined by s and θ. Performing this for all s values with given θ produces one projection. The number of projections required for effective retrieval can vary due

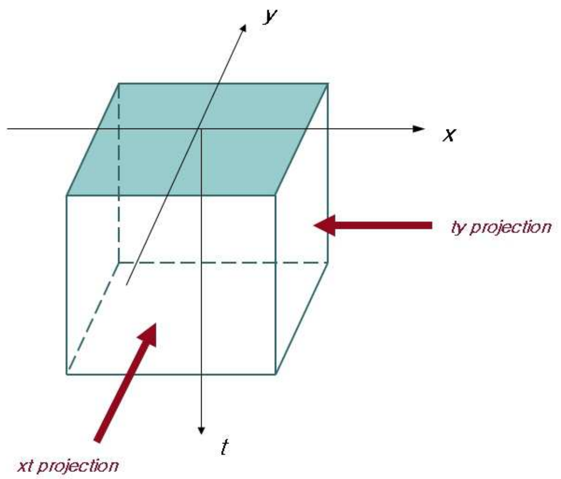


Figure 4.7: 2 Projection is performed onto video stack

to the complexity of the scene. In this thesis, we deliberately choose 2 orthogonal projections with θ equal to either 0 or $\pi/2$. Thus, the direction of the projection coincides with the image coordinate system. Each projection reduces a 2D image onto a 1D signal, which is stacked to produce 2D projection views of a 3D video stack.

The Radon transform is a very general tool to reconstruct an image from a series of projections. This defines the relationship between 2D objects and their projections. Both perspective and parallel projections can be used in the transformation. The projection function in this case is not binary. Rather this function is dependent on the density of the volume that is being projected. X-ray computed tomography (CT) obtains 3D representation of the internal structure of an object by applying the Radon transform [38]. This well-founded notion of reconstructing a signal of higher degree from a set of projections is also applied in 3D object shape matching [24].

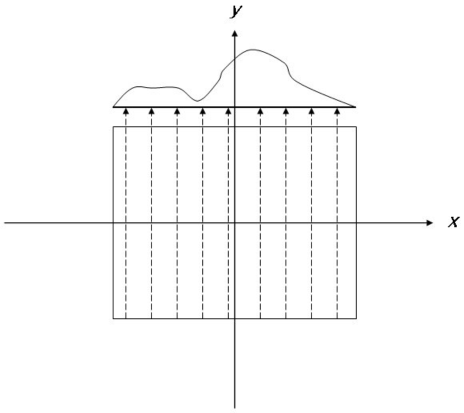


Figure 4.8: xt projection is performed with $\theta = 0$

Celenk *et al.* applied a series of transforms to video sequence to detect a simple motion of single object [17]. Motion estimation for video encoding and compression has adopted the radon projection successfully [61, 62, 6].

Our goal here is not reconstructing the 3D structure perfectly from projections. Rather we are looking for a form of representation that provides a summary of the 3D spatiotemporal stack. In the work presented here, we apply two orthogonal Radon projections to each image and stack the resulting rasters. These two resulting projection views reduce the complexity of the data significantly and open up very interesting research opportunities.

4.2.2.1 *xt* Projection

The xt projection is constructed by projecting through the stack along the y axis. Each image is projected into a single raster along the y axis, and this raster

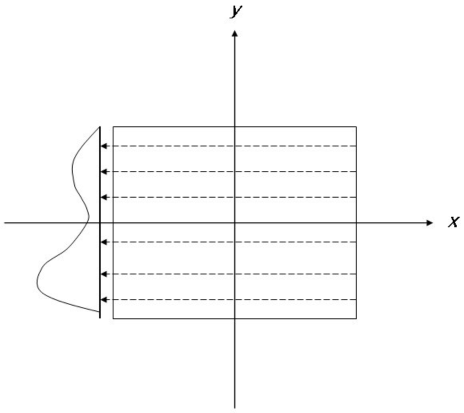


Figure 4.9: *ty* projection is performed with $\theta = \pi/2$

becomes a row of the xt view. Thus, this view captures the horizontal motion element in a video data. The dimension of the view will be $W \times T$ where T is the number of images. Equation 4.4 with $\theta = 0$ becomes

$$g\left(0, s\right) = \sum_{x=0}^{W-1} \sum_{y=0}^{H-1} f\left(x, y\right) \delta\left(x - s\right). \tag{4.5}$$

4.2.2.2 *ty* Projection

The ty projection is constructed by projecting through the stack along the x axis. Each image is projected into a single raster along the x axis, and this line becomes a row of the ty view. This view captures the vertical motion element in video data. The dimension of the view will be $H \times T$. Equation 4.4 with $\theta = \pi/2$ becomes

$$g\left(\pi/2, s\right) = \sum_{x=0}^{W-1} \sum_{y=0}^{H-1} f\left(x, y\right) \delta\left(y - s\right). \tag{4.6}$$

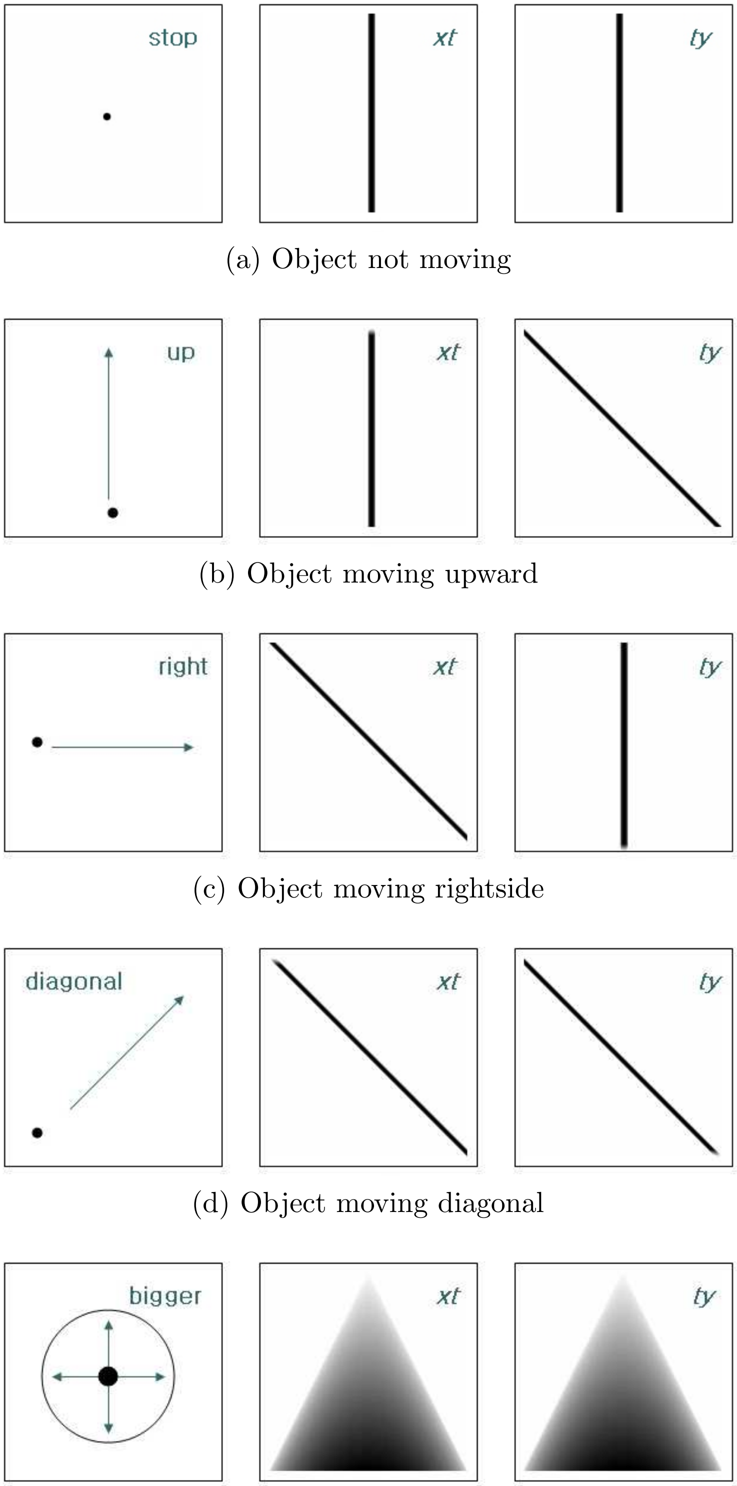
4.2.2.3 Motion profiles of projection

The projection process transforms the streaming video data onto a spatial representation. The practice of orthogonal projections and its backprojections allow understanding of the motion profile in projection data. The scope of such temporal extent may not be fine-grained, but trail-based patterns are readily seen. The projected view of spatiotemporal volume displays the following characteristics:

- Motion in video data leaves distinct tracks, providing an opportunity for event retrieval.
- The output tracks and the motion itself are correlated and predictable. In other words, it is possible to build *motion profiles*.

Figure 4.10 show the motion profile generated by our synthetic video data. Each video has a dimension of 300×300 with a duration of 300 frames. All video contains a black circle moving in a certain direction on a white background. For each figure, the first image describes the event types each video contains. The remaining two images are *xt* and *ty* projection views, respectively. Note that the vertical banding in both *xt* and *ty* projection results from lack of motion. Thus, the vertical banding in *xt* means there exists no horizontal motion within the video data. Figures 4.10d shows more complex diagonal motion which is in fact the combinations of *right* and *up* motions.

The benefit of these characteristics comes when understanding the motion-patterns within video data. Figure 4.11 features several people playing soccer. The primary figure (the person in a white shirt) wanders slowly to the left of the frame,



(a) Object not moving

(b) Object moving upward

(c) Object moving rightside

(d) Object moving diagonal

(e) Object getting bigger

Figure 4.10: Synthetic video projection showing motion profiles

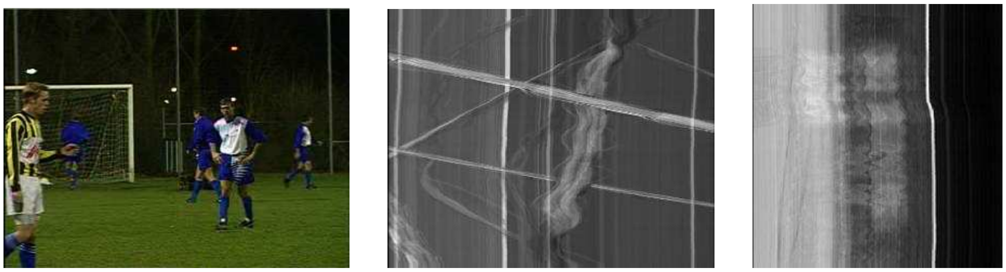


Figure 4.11: People playing soccer: A keyframe and its xt and ty projections

and this can be seen in the middle of xt view. Another primary figure (the person in a yellow shirt) appears from the left and walks off to the right, passing (in front of, since the yellow shirt trace overlays the white shirt trace) the person in a white shirt. This is also well represented in xt view.

4.3 Discussion

By taking only two orthogonal projections, the data complexity reduces significantly, but some details are lost. This is clearly shown in Figure 4.12, where the image is reconstructed by two orthogonal backprojections. The original data shows a circle, but the backprojection process can only tell the object of interest is within the square on the right. It is clear that perfect reconstruction requires more than two orthogonal projections. However, the choice of two projections is justifiable under the systematic criteria of event-based retrieval framework. The xt and ty projections present the motion summaries along the image coordinate system. Also, it is easier for human users to understand the projection views when it coincides with the image coordinate system.

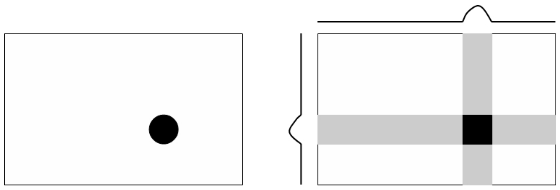


Figure 4.12: Image reconstruction with 2 orthogonal backprojections.

As we can see from Equation 4.4, the choice of $f(x, y)$ can result in different projection views. For this thesis, we choose $f(x, y)$ to be a 256-bin graylevel intensity value. Once the projection is performed onto the spatiotemporal volume, the projection data is normalized to fit into image format in grayscale (in 256 bins). This is primarily due to the concern over the size of projection data formats. Representing the projection views in textual format results in large-sized data. The image format in JPEG compresses the data well and provide the means to view the data directly. One more option is to store the rasters in a relational database, which we also did for this thesis.

Normalizing the projection data into a grayscale image with Min-Max normalization is problematic since the real maximum (and minimum) values are unknown in the real world scenario. This means that one might need to readjust the projection images later once a different set of maximum and minimum values are encountered. One viable solution is choosing predetermined minimum and maximum values and ignoring the values outside of the presets. Empirically, the projection data ranges from about 60% to 80% of the maximum.

One important issue regarding artifacts and noise inherently present within

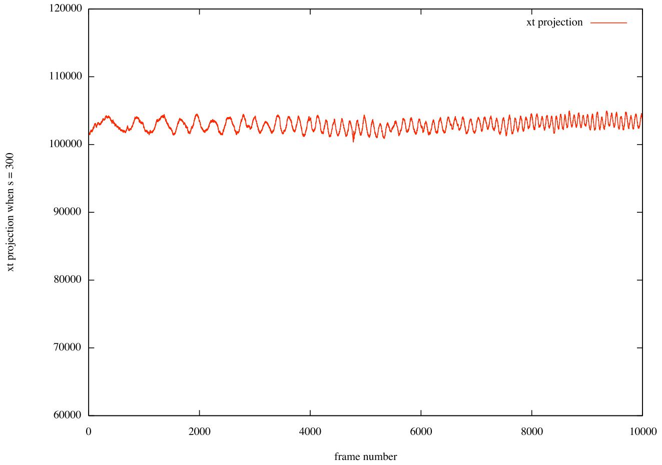


Figure 4.13: The plot of the actual xt projection of Eq. 4.5. The sample of 10k frames contains no activity but shows oscillating signal behavior. This plot is based on the single Radon projection performed when $s = 300$.

video data should be noted when considering normalization. Due to its size and complexity, the video data often requires a series of compressions and encodings. MPEG compression involves discrete cosine transform and quantization [110]. This lossy compression scheme introduces various kind of noise such as blocking artifacts and mosquito noise. Additionally, the video data is often spatially and temporally downsampled to meet reduced bandwidth. This randomly appearing noise can fluctuate the projection value and appear in the projection image.

Ideally, when there is no movement with stationary camera, each frame in a spatiotemporal volume outputs a constant projection raster. Thus, the single projection is constant throughout temporal line in both xt and ty views, resulting in vertical banding. However, the actual projection output displays a somewhat os-

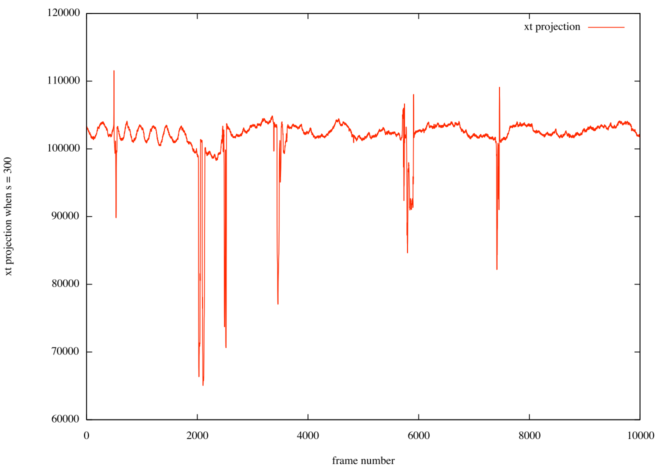


Figure 4.14: The plot of the actual xt projection of Eq. 4.5. The sample of 10k frames contains some acvitity, and the plot shows that effect. This plot is based on the single Radon projection performed when $s = 300$.

cillating behavior. As we can see from Figure 4.13, this oscillating behavior of the signal is apparent. The sample of 10k frames was obtained from the target data, where there is no activity within the sampled frames. The projection results display patterns neither constant nor random. Rather, the signal seems converging onto a higher frequency. All other samples that we have examined have displayed similar oscillating behavior.

Motion interrupts static oscillation. Figure 4.14 shows the plot of the xt projection of 10k frames when $s = 300$. Where there is motion present, the projection signal displays random fluctuation close to motion occurrence. However, we can also observe that the signal goes back to oscillation the farther away in time we are from the motion occurrence. This variable frequency activity is likely to be caused by multi-pass encoding which analyzes the input data to determine its data rate. This randomness and fluctuation were also noted by several TRECVid participants [130]. Since the retrieval system has no oversight on the data acquisition and encoding processes, robust techniques are required to overcome this inherent noise.

Selection of improper maximum and minimum normalization values can over-emphasize the fluctuating signal. For this thesis, we choose the absolute maximum and minimum value for normalization. For $W \times H$ frame of n-bin graylevel value, Equation 4.4 theoretically gives the projection raster with:

- $MAX_{xt} = nH$
- $MAX_{ty} = nW$

where MAX_{xt} and MAX_{ty} are the maximum projection values of xt and ty projections. Empirically, the fluctuation affects less than 3% of entire projection range. When the absolute max/min values are used for normalization, we found that the effect of fluctuation is minimal.

If there is no video event (i.e., no motion), the projection results in a constant raster value for each frame. The objects and the background present within video data are not discernible, and this is represented as vertical bandings in xt and ty projection views, as can be seen in Figure 4.15. Any motion generates variation away from background and results in a distinct motion track. However, the projection may not be unique among all potential motions. Figure 4.16 illustrates the case where the different set of inputs giving the identical projection results. We recognize that there are other projection functions that may deal with this type of data, but we conjecture

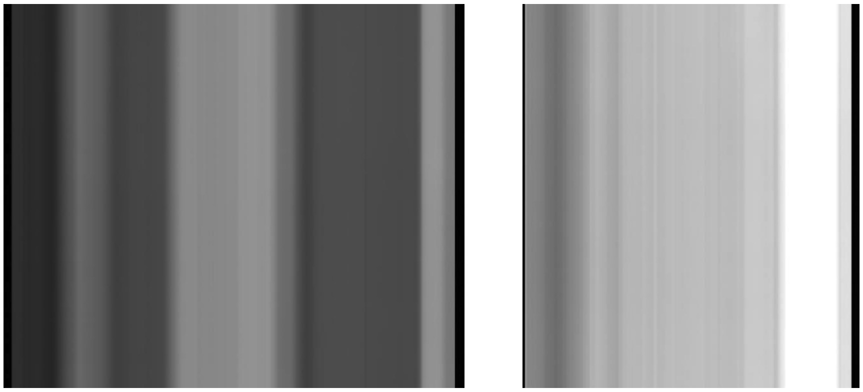


Figure 4.15: Vertical banding shown in *xt* and *ty* projection views

2	2	2	2	2	2	12
0	3	0	4	3	2	12
1	1	1	8	0	1	12

Figure 4.16: The example of Radon projection function giving the same results for different inputs.

that our choice of density projection is a logical step away from binary projection and sufficient for event detection.

The vertical banding can be removed by frequency filtering using the Fourier transform. The 2D discrete Fourier transform and its inverse transform are given by [38]:

$$F\left(u,v\right)=\sum_{x=0}^{W-1}\sum_{y=0}^{H-1}f\left(x,y\right)e^{-j2\pi\left(\frac{ux}{W}+\frac{vy}{H}\right)} \tag{4.7}$$

$$f\left(x,y\right)=\frac{1}{WH}\sum_{u=0}^{W-1}\sum_{v=0}^{H-1}F\left(u,v\right)e^{j2\pi\left(\frac{ux}{W}+\frac{vy}{H}\right)} \tag{4.8}$$

where $f(x,y)$ is an image of size $W \times H$. The transform $F(u,v)$ is evaluated for all the values of $u = 0, 1, ..., W - 1$ and $v = 0, 1, ..., H - 1$.

Generally, the Fourier transform is computationally expensive taking $O(n^2)$ operations. For more faster performance, we have taken the Fast Fourier transform (FFT) which is $O(n \log n)$. This improvement is still quite expensive in large-scale video data. In spite of this, the Fourier transform has its own desirable properties in our case. The convolution theorem states that convolution in the spatial domain corresponds to multiplication in the frequency domain. This allows the large-scale correlation operation to be more efficient. Figure 4.17 shows the results of the filtering applied on *xt* projection. The first projection view shows the normalized view with 35% of the maximum range. The second image shows the plot of the freqency spectrum after FFT is performed on the first image. The last image is the reconstruction after taking frequency filtering to remove the vertical banding.

With thresholding and filtering in the frequency domain, we have found that vertical banding can be effectively removed in both *xt* and *ty* projection data. If the task approaches projection as object recognition or trail extraction, such signal refinement can be beneficial. In the subsequent feature extraction and event detection experiments we present, we do not perform such signal processing. We extract local salient points to capture the characteristics of events and avoid any filtering process may introduce unwanted values or remove important projection information.

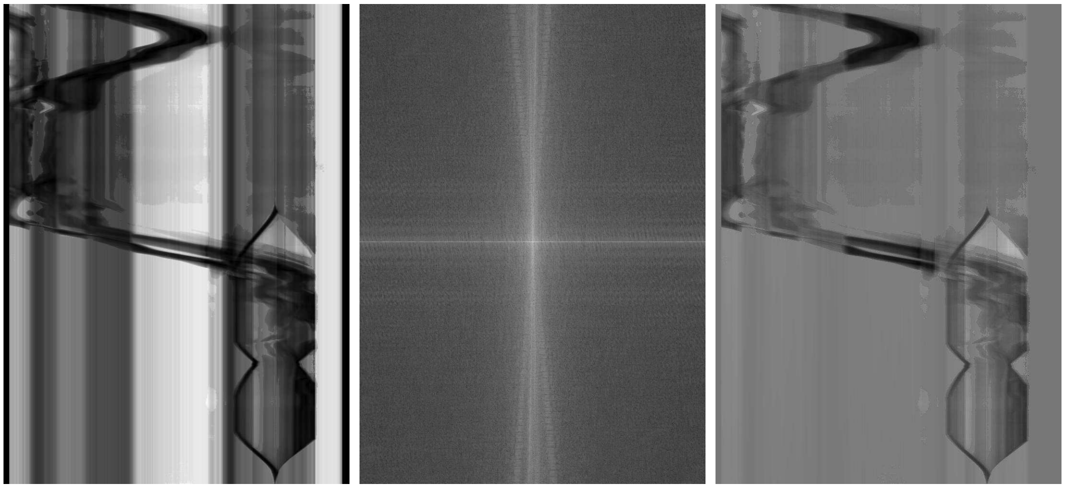


Figure 4.17: The *xt* projection: normalized with 35% range, frequency spectrum after 2D Fourier transform, reconstruction after frequency filtering

CHAPTER 5
VIDEO EVENT DETECTION

5.1 Introduction

2D information detection approaches include content-based image retrieval [102, 72, 81, 25], objection recognition [15, 138, 8] and image registration [139, 95]. The general framework begins with feature extraction using pixel-level operations. Some algorithms work solely on individual pixel values while others exploit neighborhood relationships such as edges or textures. Most content-based retrieval systems use either global, local or grid-based features. Global image features describe an image as a whole with a single vector [122]. Local features are computed at multiple interest points in the image and consequently tend to be more robust to occlusion and clutter [74]. Grid-based features can be obtained from either global or local features by using pre-determined grids in the image [14]. This accomodates geometrical relationships without extensive segmentation and object extraction.

The most common method for comparing two images is using an image distance measure. This approach retrieves nearest neighbor matches in a high dimensional feature vector space. A closer distance in the space indicates a more similar set of images [121]. Complex multimedia data requires a careful consideration for distance measures as suggested by [107, 50], but Euclidean distance remains the most popular for image similarity. When the feature distribution is summarized in histogram form, histogram intersection can be used [122]. Adopted from text document retrieval, cosine angle similarity between feature vectors has been utilized [94]. To solve various cardinalities in feature sets, pyramid match kernels are a popular approach for partial matching in feature space [39, 40]. This was extended to partial matching on spatial relationships in [65].

In our example-based approach, a target event is recognized by comparing an image and a given example. Template matching is an image registration technique in digital image processing for finding parts of an image which match a given template image. We conjecture that motion-induced projection produces similar templates and compare them with correlation. The correlation can be performed with an intensity pattern [90] or correspondence between image features. Recent advances in image understanding show promising results with local interest points. They generally are derived from gradient difference between one's spatial and scale neighbors. The seminal work by Lowe [74] introduced a local feature descriptor based on a scale invariant DoG (difference of Gaussian) operator called SIFT (Scale Invariant Feature Transform). Since then, many variants of this approach have been proposed, including SURF [7] and PCA-SIFT [58]. The performance of available local salient features are comparable [126].

5.2 Base feature extraction framework

Scale invariant feature transformation (SIFT) forms a collection of feature vectors, each of which is invariant to image translation, scaling, rotation and partially invariant to illumination changes. SIFT belongs in the category of local salient features. The algorithm is applied in a more efficient manner by taking a cascade filtering approach, in which the more expensive operations are applied only at locations that pass previous steps.

SIFT starts by building a scale-space with a series of smoothed and resampled images. The scale-space framework represents an image as a multiscale representation with a family of smoothed images [70, 3]. The notion of scale has been used extensively in computer vision community to capture the multiscale nature of real-world objects. The choice of the smoothing kernel used for suppressing fine-scale structures is the Gaussian function:

$$G\left(x, y, \sigma\right) = \frac{1}{2\pi\sigma^2} e^{-\left(x^2+y^2\right)/2\sigma^2}, \tag{5.1}$$

where σ is the size of the Gaussian kernel. The DoG (difference-of-Gaussians) function is applied, and local keypoints are extracted from local extrema. For input image $I\left(x, y\right)$, the DoG image $D\left(x, y, \sigma\right)$ is:

$$D\left(x, y, \sigma\right) = L\left(x, y, k\sigma\right) - L\left(x, y, \sigma\right), \tag{5.2}$$

where $L\left(x, y, \sigma\right)$ is the convolution of $I\left(x, y\right)$ with a variable-sized kernel Gaussian $G\left(x, y, \sigma\right)$. Thus, Equation 5.2 can be rewritten as:

$$D\left(x, y, \sigma\right) = \left(G\left(x, y, k\sigma\right) - G\left(x, y, \sigma\right)\right) * I\left(x, y\right), \tag{5.3}$$

where $*$ indicates the convolution operation. By taking the difference of two Gaussian-blurred images with different σ, the algorithm selects the candidate interest points by identifying local maxima/minima in scale-space.

The extremity detection process generates too many keypoints, and the algorithm prunes according to its contrast and edge response criteria. After a series of localization processes, one (or more) of the main magnitude and orientation for keypoint are assigned based on local image gradient directions from DoG image pyramids. The subsequent keypoint descriptor becomes rotation-invariant as the keypoint descriptor can be represented relative to this orientation.

The last step in the algorithm computes a descriptor vector for each keypoint. The SIFT feature generates four parts: location, scale, orientation and descriptor. The descriptor vector is formed over a 16×16 region around the key point. The region is divided into 4×4 subregions, and an 8-bin orientation histogram is created for each subregion. Thus, the descriptor is a 128-element feature vector for each interest point.

Once the descriptor is extracted, the images can be compared by calculating the distance between the descriptors. The feature vectors in closer proximity are considered a match. The nearest neighbors can be defined as the keypoints with minimum Euclidean distance from the given descriptor vector. However, this approach can prove to be too computationally expensive when the number of SIFT points is extremely high. Also, this number tends to be quite high for typically-sized images. It is not uncommon to see a couple thousand interest points for 500×500 images. This feature can also be used as grid descriptor [33]. In this case, the local keypoint detection process is skipped, and the descriptor is directly formed over regular 16×16 grids. Figure 5.1 shows the keypoints detected for a sample xt projection from camera

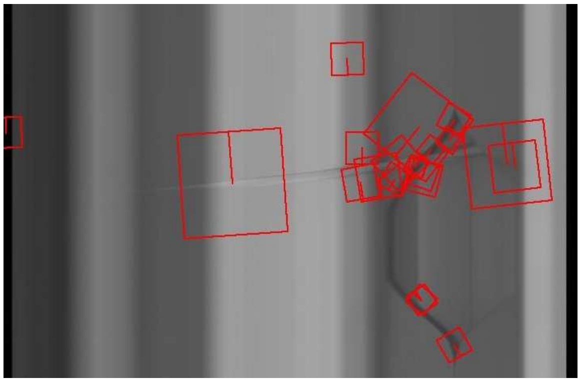


Figure 5.1: SIFT keypoints for camera 4. This shows elevator door opening and its corresponding SIFT keypoints. Scale and orientation are shown as the size of rectangle and its direction.

4, where the segment shows an elevator door opening and closing.

5.3 Matching strategy

When the projection gives a distinct event pattern present within a spatiotemporal volume, the task of event detection becomes finding a *similar looking* pattern within a target video projection. This can be considered to be the task of geometrically aligning event patterns between projections. Procedures for mapping points from the reference projection to corresponding points in the target projection are found in image registration techniques [139]. Image registration typically involves model transformation and estimation to compute the mapping functions between images. In our problem, we consider the task of temporal mapping of events. This point of alignment is found along the temporal extent, using a sliding window.

Cross correlation is a popular technique to find known signal features in a long duration of signal, measuring the signal similarity using a sliding dot product. In image correlation, this usually involves sliding the reference image $r\left(x,y\right)$ along the target image $t\left(u,v\right)$. The *sum of abolute differences*(SAD) is the simplest way of computing the correlation [128]:

$$c\left(x,y\right)=\sum_{x,y}\left|r\left(x,y\right)-t\left(u+x,v+y\right)\right|, \tag{5.4}$$

where the sum is over x, y under the window containing the feature t positioned at u, v. We used this correlation method for trail-based event retrieval in [90]. More complex methods normalize the image features due to brightness variation [68]. The *normalized cross correlation* uses the convolution of $r\left(x,y\right)$ and $t\left(u,v\right)$:

$$c\left(x,y\right)=\frac{\sum_{x,y}\left\{r\left(x,y\right)-\bar{r}\right\}\left\{t\left(x-u,y-v\right)-\bar{t}\right\}}{\left\{\sum_{x,y}\left\{r\left(x,y\right)-\bar{r}\right\}^{2}\sum_{x,y}\left\{t\left(x-u,y-v\right)-\bar{t}\right\}^{2}\right\}^{\frac{1}{2}}}, \tag{5.5}$$

where the convolution process takes $t\left(-u,-v\right)$ and normalizes to produce cosine similarity-like correlation. Another popular method is to perform correlation in the frequency domain [97]. Due to the convolution theorem, convolution in the spatial domain corresponds to multiplication in the frequency domain. This allows the sliding dot process to be more efficient in the frequency domain. However, even with a FFT approach, we found that the transformation alone is too expensive for large-scale data.

The problems we found with the intensity-based correlation approach in event detection are generally twofold:

- The corresponding points between event patterns tend to match in trail-based events. If the scope of an event feature requires more than comparison of motion paths, correlation methods simply do not have enough feature refinement to capture locally salient descriptions.

- Even with trail-based events, we find that the general events require a high level of variation. Additionally, the highly variable signal is naturally highly susceptible to false alarms.

Feature-based similarity is computed over the set of features extracted from the reference projection and the target projection. Due to the event variation issue, the task at hand requires a more flexible approach than the exact feature point matching such as that found in [74]. A histogram is an estimating representation of the distribution of a variable. In image processing, it is used to effectively perform density estimation and comparison of the distribution of the underlying feature variable such as colors. Histograms provide a compact summarization without strict object segmentation and are well suited for the problem of recognizing an object of unknown position within a scene. Research utilizing histograms for such tasks can be found as far back as 1991 where color histograms were used [122]. They employed a global histogram framework for real-time image object classification. More recently, a histogram was utilized for a gradient-based feature for the task of human detection [22]. They showed that a locally normalized HoG (histogram of gradients) can perform comparably with other local interest algorithms when it is applied to dense image grids for visual object recognition. Histograms are compared by calculating similarity using histogram intersection. Given a pair of histograms, P and R from projection and reference respectively, the normalized histogram intersection I is:

$$I\left(P,R\right)=\frac{\sum_{i=0}^{n} min\left(P_i,R_i\right)}{\sum_{i=0}^{n} R_i}, \tag{5.6}$$

where n is the number of bins for each histogram.

Another way of calculating the similarity of feature vectors is cosine similarity [106, 94], which is heavily used in document retrieval and classification. In the term vector model, text documents are represented as vectors of index term frequencies. Typically, terms refer to keywords extracted from documents. Direct distance (i.e. Euclidean distance) in vector space can be problematic since the magnitude of a term vector is impacted by the length of the document. For instance, a longer document has higher chance of producing more distinct terms and term occurrences, resulting in a larger magnitude. To compensate for this issue, similarity is often measured as an angle between two term vectors. Similarity for image retrieval is calculated in a similar manner. For feature vectors, v_A and v_B, obtained from two images I_A and I_B, the cosine similarity is given by:

$$cosinesimilarity\left(v_A,v_B\right)=cos\theta=\frac{v_A\cdot v_B}{|v_A||v_B|}, \tag{5.7}$$

where a smaller θ makes the cosine of the angle approach one, meaning the two images are more similar.

Recent work in object-based image retrieval in large-scale corpora has adopted simple text-retrieval techniques using the analogy of a *bag of words* [114, 86, 92]. This is also actively investigated in scene categorization [33, 132]. The BoW model comes from natural language processing where a text document is represented as an unordered collection of words. For instance, latent semantic analysis uses the BoW model to analyze the relationship between a set of text documents via conceptual clustering based on semantic structure [29].

In computer vision and image processing, an image can be treated as a document and is represented as a set of unordered basic features (the words or codewords). The typical words used in image processing are local image features such as SIFT or image grid descriptors. A feature characterizing either the salient point or the region is computed. The resulting features are represented in high-dimensional space and are categorized using quantization techniques such as k-means. This process outputs the dictionary of the object categories. The number of vocabularies within the dictionary is an important experimental setting as too few words may not be descriptive enough for classification. Each object is defined by a set of words (or a bag of words) from the dictionary, and detection is based on the frequencies of the words in codebook.

This approach is especially desirable in event detection on large-scale corpus since (1) geometrical relationships are deliberately ignored and (2) the features are quantized to fit into the visual vocabulary. Building a vocabulary is hence not an event-specific task and is performed as general indexing. Once the code generation is performed, the similarity can be easily computed with either histogram or cosine similarity. The selection of vocabulary size can be an important design issue. Research by the vision community indicates that about a couple hundred codewords for image classification is typical [65].

Each input example e forms an example codebook histogram H_e with its number of bins is equal to the number of vocabularies in the codebook. From the set of examples $E = e_1, e_2, ..., e_n$ for event type E, we get an event representation defined by the mean and variance of all the corresponding examples. Since each event example has different duration, each example histogram is normalized to the unit duration. The mean and variance are computed for each example histogram bin separately. The mean histogram H_M is the event representation given by multiple examples, and the variance histogram H_V represents the reliability of the individual histogram bins. If a certain codeword bin has high variance, this indicates that the codeword has low significance in event representation. Thus, our similarity score based on histogram intersection between event definition E and input video projection P is the sum of two separate histogram intersection in xt and ty projections:

$$\begin{aligned} similarity\,(P, E) &= \mathcal{I}\,(H_P, H_E) \\ &= \alpha\mathcal{I}\,(H_{Pxt}, H_{Ext}) + \beta\mathcal{I}\,(H_{Pty}, H_{Ety})\,, \end{aligned} \tag{5.8}$$

where the histogram intersection in Equation 5.6 is weighted with $\omega_i = \frac{1}{H_V(i)}$:

$$\mathcal{I}\,(H_A, H_B) = \sum_{i=1}^{n} \omega_i \cdot min\,(H_A\,(i)\,, H_B\,(i))\,. \tag{5.9}$$

One useful extension for using histograms in event detection is the use of a multi-resolution histogram [42]. Particularly, in the bag-of-words approach with SIFT keypoints, we use the *pyramid matching* algorithm proposed in [39, 40]. Pyramid matching finds the correspondence between two feature vectors by placing a sequence of increasingly coarser grids over the feature space. At each resolution, the algorithm takes a weighted histogram intersection among points that did not match at a finer resolution. A higher weight is given for matches at a finer resolution. For a given resolution l, the histogram intersection between E and P is $\mathcal{I}\left(H_P^{(l)}, H_E^{(l)}\right)$. Then, the similarity based on the pyramid match kernel $k \vartriangle$ is defined as:

$$\begin{aligned} similarity\,(P, E) &= k \vartriangle (P, E) \\ &= \sum_{j=0}^{L} \frac{1}{2^j} N_j, \end{aligned} \tag{5.10}$$

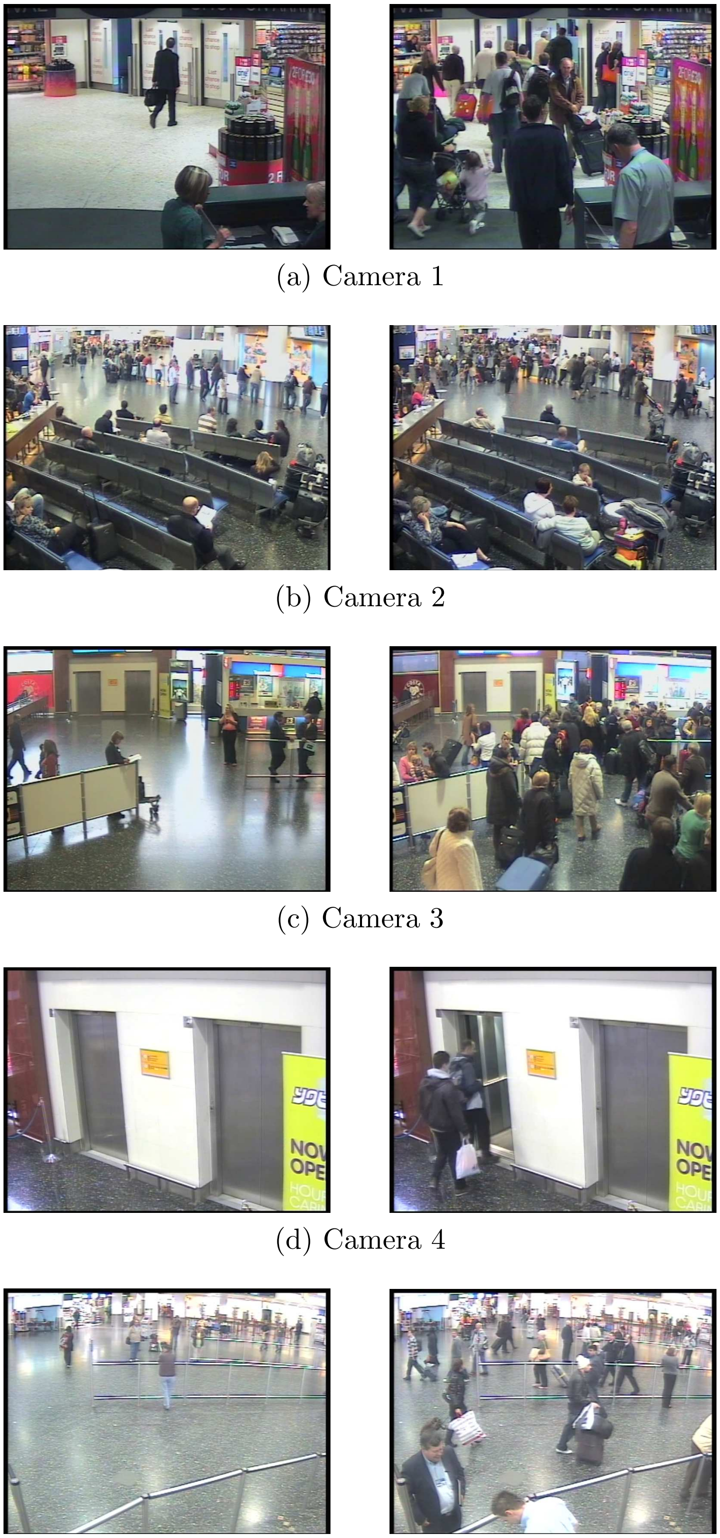
where N_j is the number of newly matched points at level j, which did not have any matching at finer resolution levels. N_j is calculated by the histogram intersection at level j:

$$N_j = \mathcal{I}\left(H_P^{(j)}, H_E^{(j)}\right) - \mathcal{I}\left(H_P^{(j-1)}, H_E^{(j-1)}\right), \tag{5.11}$$

where the intersection operator $\mathcal{I}$ is based on Equation 5.9.

5.4 Dataset

Our dataset comes from surveillance video recorded over 10 days at London Gatwick airport. It consists of 5 different fixed cameras, each in a different location, with different backgrounds and traffic trends. The data comes as 50 segments of surveillance video data (10 days * 2 hours/day * 5 cameras). The 100-hour corpus is divided into training and testing sets, each comprising 50 hours of the corpus. The total size of the corpus is about 250GB or 2.5GB/hour. This is fairly large considering its resolution, which is 720×526 at 25 frames per second. The original source format is unknown, but the corpus files are in MPEG-2 format. Video compression artifacts and noise are visible throughout the dataset, as it is inherently present in MPEG coding.



(a) Camera 1

(b) Camera 2

(c) Camera 3

(d) Camera 4

(e) Camera 5

Figure 5.2: Examples of each camera location

Figure 5.2 shows some example keyframes from each camera location 1 to 5. Camera 1 (Figure 5.2a) shows a door, for which the normal flow of traffic is defined to be going out. The visible security usually stays around the bottom right corner in the frame. People walk through the door, and occasionally a cargo truck drives through. Camera 2 (Figure 5.2b) shows the waiting area with a series of long chairs. The people in the far background can be challenging to recognize given their small relative size, and the traffic can be very heavy at times. The top right corner is what is being seen in camera 1, which can be barely recognizable with the given resolution. Camera 3 (Figure 5.2c) shows people waiting outside fences. In the background, we see the elevator doors used in camera 4. This location is what is shown in the far background in camera 2. People typically enter the scene from the bottom of the frame. Camera 4 (Figure 5.2d) shows two elevator doors with people coming in/out of them. This is the scene with the lowest level of traffic. Camera 5 (Figure 5.2e) shows people moving primarily left/right with fences dividing the scene. This view is unrelated to any other camera location. The traffic can be heavy at times. The data not only have encoding artifacts, but also there are some surfaces with shining, blinking and reflecting properties, as well as various active video monitors presenting information. This can be best seen in one keyframe in Figure 5.2c. The evaluation framework we used is limited to a single-camera approach where each camera view is processed independently.

5.5 Events and annotation

A video event denotes an observable action in a video stream. The set of events to be detected is:

- *CellToEar*: someone puts a cell phone to ear
- *ElevatorNoEntry*: elevator door opens and a person does not get in
- *Embrace*: someone wraps one or both arms around another person
- *ObjectPut*: someone puts down an object
- *OpposingFlow*: someone moves through a door against a normal flow of traffic
- *PeopleMeet*: people move to each other and communicate
- *PeopleSplitUp*: people separate from a group and leave
- *PersonRuns*: someone runs
- *Pointing*: someone points
- *TakePicture*: someone takes a picture.

More information about event definition as used here can be found in [111]. The event annotation is provided by The Linguistic Data Consortium and the National Institute of Standards and Technology.

As discussed in [101], the annotation of real-world video data of this size proved to be very challenging and time-consuming. The event annotation typically takes $10 - 15\times$ actual video time. While the event description is well understood by human annotators, the actual annotation often can introduce the event ambiguity - i.e., *Pointing* or gesturing? is a baby an object in *ObjectPut*? The annotation denotes event type, starting frame and ending frame. This introduces the temporal ambiguity in event definition. For instance, if a person holds a cellphone by his ear for a long time, is it *CellToEar* when there is no cellphone to ear motion? When should one declare the ending frame for such event? The annotation ambiguity coupled with human fatigue results in incomplete and incorrect annotation.

Even though the video event takes more than a single frame to be detected, the actual annotation contains many instances with zero duration (i.e., the starting frame is equal to the ending frame). Table 5.1 shows *CellToEar*, *Embrace*, *ObjectPut*, *PeopleMeet*, *PersonRuns* and *Pointing* have a minimum duration of zero. The number of annotation items with zero duration is low, and we disregard such items from our evaluation when extracting each event definition. This is because our system is limited to the temporal event where there is an observable change of state. This requires the minimum event duration of 2 frames. We can also see that some events have very high temporal variation - *CellToEar*, *Embrace*, *ObjectPut*, *PeopleMeet*, *PeopleSplitUp* and *Pointing*. These events have very high maximum duration numbers compared to their average duration.

Table 5.1: Event average, maximum and minimum duration in training set

	Average Duration	Maximum	Minimum
CellToEar	40	4745	0
ElevatorNoEntry	359	467	297
Embrace	157	3188	0
ObjectPut	21	538	0
OpposingFlow	60	234	12
PeopleMeet	110	1674	0
PeopleSplitUp	232	3392	0
PersonRuns	76	386	0
Pointing	38	1029	0
TakePicture	287	472	194

The event frequencies vary as we can see from Figure 5.3, exhibiting the real-world scenario where some events are encountered more than others. The selected events represent challenging research tasks involving various types of actions such as macro (*ElevatorNoEntry*, *OpposingFlow*) vs.micro (*CellToEar*, *Pointing*) or single actor (*PersonRuns*) vs.multiple actors (*Embrace*, *PeopleSplitUp*). Any instance of these events pose difficult vision problems.

As we can see in Figure 5.3, some events such as *ObjectPut* and *Pointing* have somewhat high frequency differences between training and testing set. If we break the event frequency into camera location (Table 5.2), we find the frequency disparity may pose problems for the detection system. This is because some events are not defined for a camera location in training but have one or more occurrences in the testing set (*Embrace* cam4, *TakePicture* cam3 and *TakePicture* cam5).

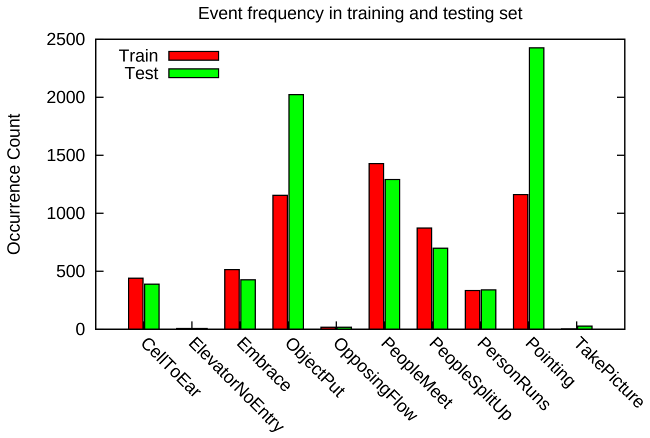


Figure 5.3: The event occurrences in training and testing set.

Table 5.2: Event frequency by camera location in training/testing set

	CAM1	CAM2	CAM3	CAM4	CAM5
CellToEar	28/12	130/133	148/135	2/0	131/108
ElevatorNoEntry	0/0	0/0	1/3	5/3	0/0
Embrace	24/3	122/94	332/291	0/2	29/36
ObjectPut	285/392	352/759	289/602	10/5	156/265
OpposingFlow	17/17	0/0	0/0	0/0	0/0
PeopleMeet	398/392	322/216	442/462	5/2	213/219
PeopleSplitUp	430/330	215/122	100/132	4/4	115/111
PersonRuns	17/8	114/112	104/114	3/1	95/104
Pointing	388/533	430/566	412/691	10/8	416/628
TakePicture	0/0	3/18	0/6	0/0	0/3

The annotation is temporally-oriented, meaning the observed events are annotated temporally. However, no spatial annotation is provided. If an event is observed, it is not known where in the frame that particular event is observed. This presents a particular challenge to the typical vision methodology, since the annotated temporal location may exhibit multiple event occurrences. The features extracted from an event segment can easily confuse the learning process when especially one-vs-all classification is used.

5.6 Evaluation

The evaluation of the TRECVid event detection task is based on another similar task, TREC spoken term detection evaluation [34]. Since the video has no temporal boundaries, the unit of detection needs to be determined by the system. The evaluation procedure needs to consider if the unit of detection is appropriate with given annotation level. However, our framework detection systems answer the question: "Is this instance of data similar to a set of examples from training data?" Each time a system answers the question, the system is using training examples as its unit of detection. This is similar to image retrieval where the unit of retrieval is an image. We are explicitly using as our unit of detection what is provided in the training annotation. Since our framework is detecting strong temporal variation within event annotation, we only attempt to detect with a *priori* knowledge of the detection window given by the training annotation. For instance, we cannot ask the system to detect how long a given *CellToEar* event lasts once it happened since its temporal variation has already happened. Thus, the TRECVid guidelines describe how the output of a system maps to the reference annotation based on their temporal alignment. Due to the way our system works with a sliding detection window, we do not consider the issue of event alignment. We do not worry about the exact temporal location of an event but rather assume that an event is present somewhere within the detection window when the similarity is within certain threshold.

System performance is graphically assessed with a Detection Error Tradeoff (DET) curve [77] as shown in Figure 5.9 through Figure 5.18. A plot is generated for a series of missed detection probabilities and false alarms that are a function of a detection threshold. Higher performance systems will plot toward the lower left corner with lower false alarm and missed detection probability. The formulas for P_{miss} and R_{FA} are:

$$P_{miss} = N_{miss}/N_{target}, \tag{5.12}$$

$$R_{FA} = N_{FA}/N_{source}, \tag{5.13}$$

where N_{miss} is the number of missed detections, N_{target} is the number of total event observations, N_{FA} is the number of false alarms, and N_{source} is the total duration of data in hours. Once the DET curve is plotted, the system performance is compared using a composite metric of the misses and false alarms. In this work, we adopt another similar detection evaluation framework [78] and compute the detection cost. Normal Detection Cost Rate (NDCR) is defined as:

$$NDCR = P_{miss} + \beta R_{FA}, \tag{5.14}$$

where

$$\beta = \frac{Cost_{FA}}{Cost_{Miss} \times R_{Target}}$$

$$Cost_{Miss} = 10; Cost_{FA} = 1; R_{Target} = 20/hour.$$

The constants were chosen as a result of discussions with the research and user communities. R_{Target} is a constant for the *a priori* event observation rate, which was arbitrarily selected to be in the middle of the event distributions in Figure 5.3. One thing to note here is the choice of cost constants for both miss rate and false alarm. $Cost_{Miss}$ is a constant for the missed observation cost, and $Cost_{FA}$ is a constant for false alarm cost. By choosing 10-times higher $Cost_{Miss}$, the evaluation is emphasizing the ability to detect more correct instances at the cost of declaring higher numbers of false alarms. Thus, the final NDCR is impacted more by P_{miss} than by R_{FA}. $NDCR = 0$ indicates the performance of a perfect system, and $NDCR = 1$ is the cost of a system with no output. The range of NDCR is $[0, \infty]$, and $NDCR = \infty$ when $R_{FA} = \infty$.

5.7 Experimental results

We use SIFT as our local salient point detector. We perform Radon projection on the entire corpus and save the resulting projections as 256-bin gray-scale images. The projections reduce the set to 600MB, which is about 0.2% of the original data. SIFT keypoints are extracted from projection images. The total number of SIFT keypoints is about four million (i.e., 3.2 million for *xt* and 0.8 million for *ty*). We perform k-means clustering on randomly sampled subsets of interest points to form a visual codebook. For this experiment, we randomly sampled 5k points for cluster definition (i.e., 1k interest points randomly selected for each camera location). We perform this process in *xt* and *ty* domain separately. Thus, the total number of

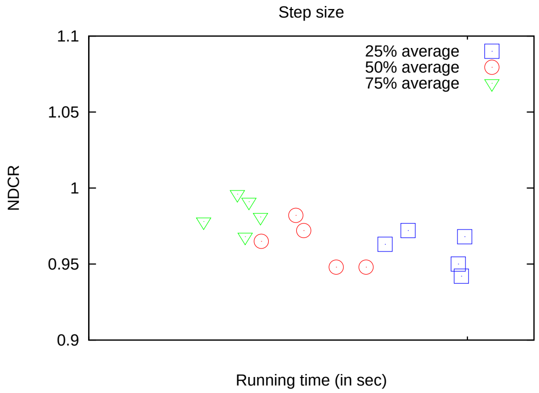


Figure 5.4: Performance and computation time based on step size

randomly selected keypoints is 10k. Clustering is performed based on the spatial location (either in x or y) and the SIFT descriptor vector. K-means clustering is performed with the Weka package [43]. Based on clustering output, we built the codebook with vocabulary size up to 2k (i.e, up to 1k for xt and ty, respectively).

Since using the sliding window with every projection row is prohibitive with large-scale data, we calculate the average event duration and take a half of the average duration as our sliding window step size. With training data, we experimented with various step sizes but saw little performance increase with finer stepping size while computation time increased (Figure 5.4). This is partly due to the fact that the smaller stepping size tends to increase the number of false alarms.

Our similarity equations (Equation 5.8 and 5.10) have a set of parameters that can be adjusted.

- Vocabulary size: This corresponds to the number of histogram bins in BOW representation. We tried from 32 bins up to 1024 bins in each of the xt and ty

projections. This parameter is not needed when pyramid matching (Equation 5.10) is used.

- xt and ty ratio: This determines how much impact each of the xt and ty projections will have on the final similarity. We tried the five settings of $[\alpha : \beta]$: $[1.0 : 0.0]$, $[0.75 : 0.25]$, $[0.5 : 0.5]$, $[0.25 : 0.75]$ and $[0.0 : 1.0]$.
- Histogram bin weights: $\omega_i = \frac{1}{H_V(i)}$ and $\omega_i = 1$.

Our evaluation was two-step process. In the first step, we performed a series of experiments varying parameter values on the training set. Based on the results from the training set, we chose the optimal settings for use with the testing set. The number of the official runs for each event was limited to three runs. In TRECVid event detection task, participants were allowed to submit multiple runs for each event. In the second step, we performed a series of runs as with the testing set to present the results from those settings that were not a part of the official runs.

Table 5.3 shows our official run results and the best runs of 2008 TRECVid participants [101]. Note that TRECVid values in this table are minimum NDCRs, which are generated from computing the optimum threshold by analyzing the DET curve. This is often considerably lower than the actual NDCR values (that is, the value resulting from the submitting thresholding value for a given system). Also, note that the parameter settings in each run may differ between events. The detailed description of the parameter settings and the DET plots from our official runs are provided in Subsection 5.7.1. All the results shown in Subsection 5.7.1 have the actual NDCR values, and this includes the best TRECVid result. We can see that the micro-scale events such as *CellToEar*, *ObjectPut* and *Pointing* are generally very hard for all systems. From the TRECVid average values, we can see that many systems struggle to break the 1.0 NDCR barrier for many events. The number of TRECVid runs for each event ranges from 4 (*CellToEar*) to 14 (*PersonRuns*). It is hard to directly compare the performance between systems, given the aggregated nature of reported system performance. However, we claim through these results that our system clearly demonstrates its capability to process a wide range of challenging events with simple detection-via-similarity approach.

The *ElevatorNoEntry* result in Figure 5.10 indicates that the system is not being able to detect one third of the instances in the testing set. This is due to limitation that the system only processes 2-hour in each segment from the dataset of 50 segments (i.e., the dataset has about 100-hour data). However, each segment may have a few minutes of data after the initial 2-hour mark. In this case, our system is not detecting the instances happening in those timeframes. In case of *TakePicture* in Figure 5.18, the system is intentionally avoiding those camera locations where the example was not provided in the training set. Thus, about one third of the event instances in the testing set remains undetected with a high false alarm rate.

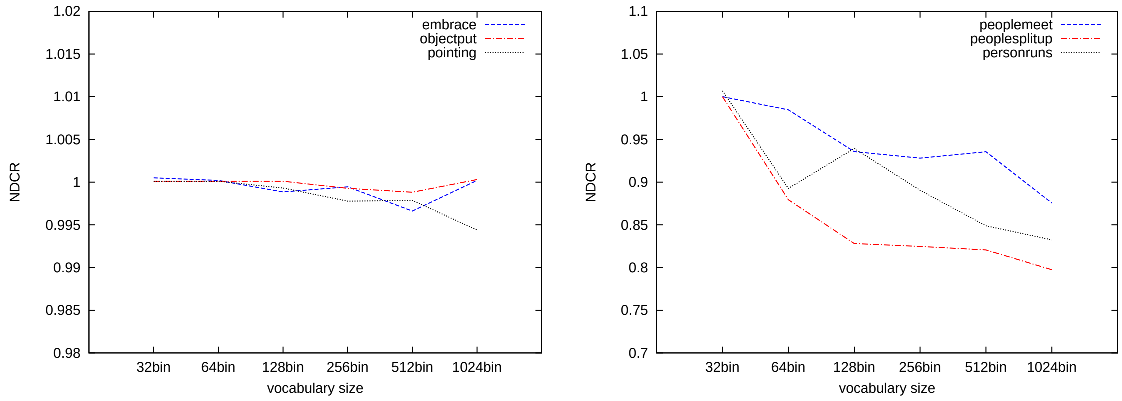
Note here that only small number of participants have multiple best runs - i.e., participant *A* (*CellToEar*, *ObjectPut*, *Pointing*), participant *B* (*OpposingFlow*, *ElevatorNoEntry*, *TakePicture*) and participant *C* (*PeopleMeet*, *PeopleSplitUp*). Participant *A* fused several machine learning algorithms and applied them on probabilistic models of human pose and motion. Participant *B* is a commercial vision firm, which heavily utilized a human detector and tracker. Participant C employed SVM techniques on an optical flow feature, only focusing on macro-type events. Across the complete range of events, we produce highly competitive performance, proving our detection-via-similarity approach on spatiotemporal projection is quite robust.

Table 5.3: NDCR comparison between our approach (run1,2,3) and TRECVid

	Run 1	Run 2	Run 3	TRECVid best	TRECVid avg
CellToEar	0.955	0.956	0.969	0.997	1.018
ElevatorNoEntry	0.373	0.377	0.400	0.0003	0.719
Embrace	0.902	0.951	0.973	0.990	1.013
ObjectPut	0.999	0.999	1.000	0.999	1.133
OpposingFlow	0.353	0.377	0.395	0.354	0.790
PeopleMeet	0.773	0.876	0.879	0.998	1.003
PeopleSplitUp	0.716	0.797	0.797	0.973	0.994
PersonRuns	0.832	0.837	0.844	0.851	1.000
Pointing	0.994	0.994	0.996	1.000	1.061
TakePicture	0.413	0.419	0.447	0.852	0.955

In the second stage of evaluation, we varied the parameter settings on the testing set. The variations we tried include vocabulary size, xt/ty ratio and histogram bin weight scheme. The goal of this process is to show the performances based on various input settings.

Based on Equation 5.8 with histogram intersection Equation 5.9, we varied the number of vocabularity size n. The general notion here is that the smaller vocabulary size favors a lower false alarm rate while the bigger codebook had better missed probabilities with lower numbers of false alarms. Figure 5.5 shows the NDCR plots



(a) (b)

Figure 5.5: NDCR plot on varying vocabulary size n. Note that two plots are on different NDCR range to illustrate each plot in detail.

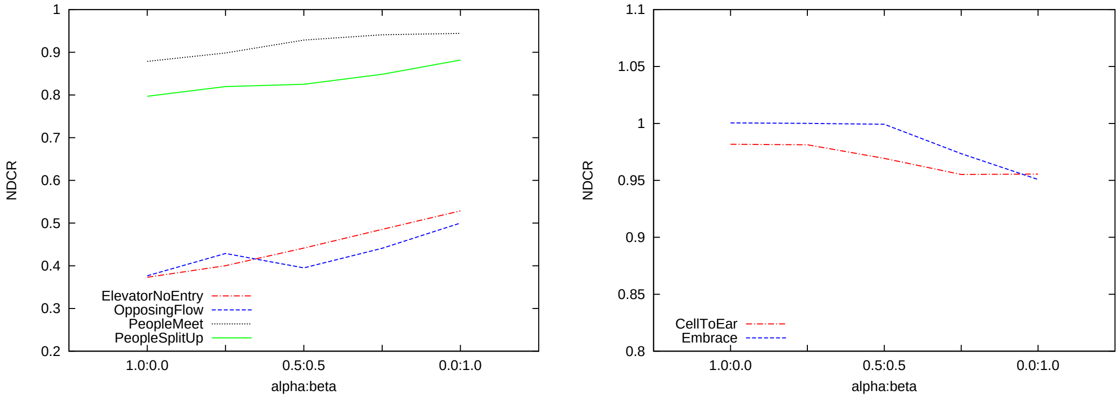
based on the vocabulary size. All runs here were performed with $\alpha = 0.5$, $\beta = 0.5$ and variation weighting. Figure 5.5a shows micro events such as *Embrace*, *ObjectPut* or *Pointing* all have no significant performance advantage with vocabulary size. This probably indicates that there are not enough features capturing the characteristics of the micro-level events. Figure 5.5b shows macro events such as *PeopleMeet*, *PeopleSplitUp* or *PersonRuns* have better performance with larger vocabulary size. This indicates that human motion-induced trajectories are well captured by projection and matching function.

With the similarity function 5.10, there is no need to consider vocabulary size for distinguishing between events. We also found that pyramid matching generally outperforms any variation in vocabulary size. However, the data size increases considerably since pyramid matching is implemented with a perfect binary tree with

k-means clustering in each node ($k = 2$). With vocabulary size n, the system needs to manage $2n - 1$ cluster definitions. To avoid this, pyramid matching approaches often use top-down or divisive clustering. In our case, we divide the clustering until $n = 1024$, which results in a perfect binary tree with a total number of levels $l = 10$ or 2047 cluster definitions.

Based on Equation 5.9, we varied α and β values to see the effect of each xt and ty projections. Even though the xt projection generally has a stronger trajectory with more keypoints, we found that some events produce better performance with ty projection. Figure 5.6 shows the performance variations when varying α and β are used. All runs here were performed with pyramid matching with variation weighting. The events that tend to have strong trajectory information seem work well with the xt projection. The micro events such as *CellToEar* and *Embrace* have better results with the ty projection. The improvement may not be significant, but their results are valuable since these micro events are considered very hard. Especially for *Embrace*, any improvement from $NDCR = 1.0005$ is important since $NDCR = 1$ is equal to the system with no output. With $[\alpha : \beta]$ being $[0.5 : 0.5]$, the system seems to respond well to a variety of events. One thing to note here is that the ty projection may not have strongly human-discernible patterns, but the projection and its feature extraction are capturing some event-relevant information. In most cases, we see the ty-only runs are not far behind those with xt-only in terms of NDCR metric.

We also tried the variations $\omega_i = \frac{1}{H_V(i)}$ (variation weight) and $\omega_i = 1$ (no variation weight). Figure 5.7 shows the NDCR performances in either case. All runs here



(a) (b)

Figure 5.6: NDCR plot on varying α and β. Note that two plots are on different NDCR range to illustrate each plot in detail.

were performed with pyramid matching with $\alpha = 0.5$ and $\beta = 0.5$. *ElevatorNoEntry*, *OpposingFlow* and *PersonRuns* all had better NDCR with $\omega_i = \frac{1}{H_V(i)}$. At least in the case of *ElevatorNoEntry* and *OpposingFlow*, this indicates that the projection and subsequent feature extraction are capturing the relevant information within the detection window. When other irrelevent keypoints are suppressed by variation weighting, this leads to higher performance. However, *Embrace*, *PeopleMeet*, *PeopleSplitUp* and *TakePicture* had better performance with no variation weight. This indicates that a high level of variation can exist within the same category of event examples. This is especially true in the macro type events such as *PeopleMeet* and *PeopleSplitUp*. Their characteristics are strongly derived from motion trails and their interaction, which tend to be highly variable.

Figure 5.8 shows the actual xt projection views from *CellToEar*, *Embrace*,

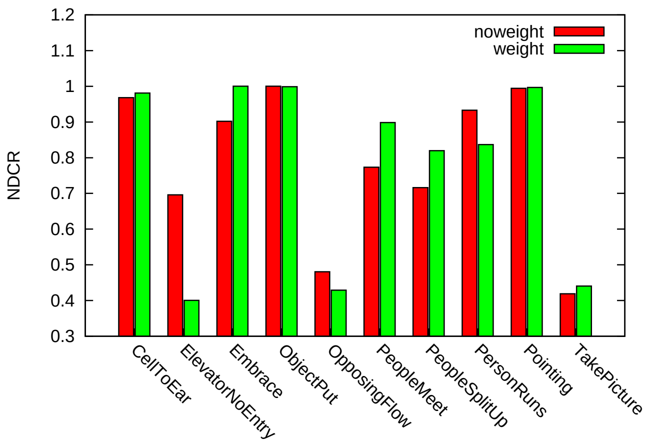
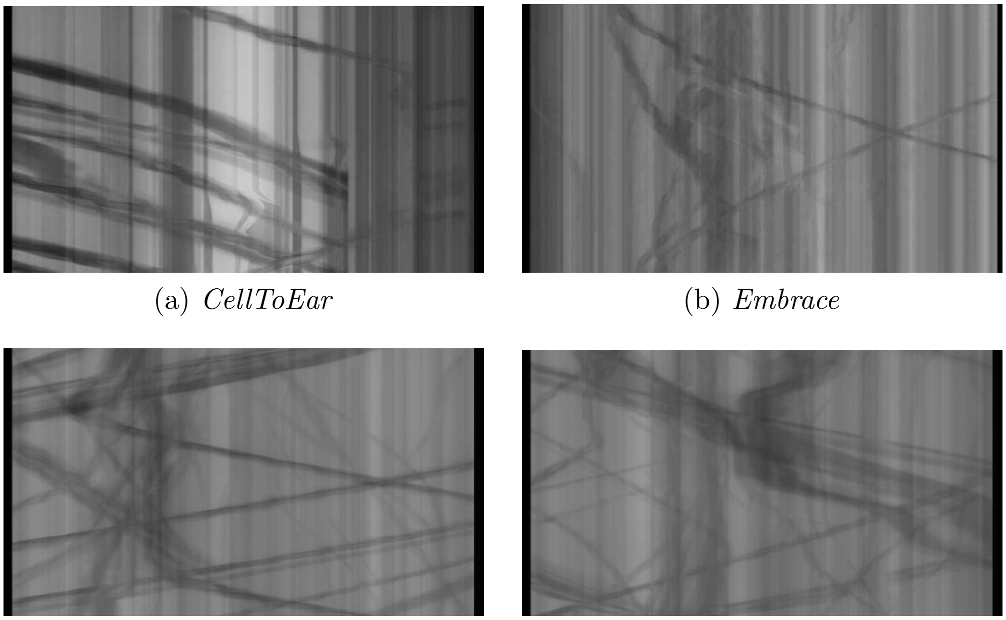


Figure 5.7: NDCR plot when either weight schem is used. Note that the lower NDCR indicates better performance.

PeopleSplitUp and *PersonRuns*. The micro events such as *CellToEar* (Figure 5.8a) might be impossible to recognize by a human user, but some macro events such as *PeopleSplitup* in Figure 5.8c may be recognizable. With global histograms extracted from the area under a sliding window, a high number of irrelevant keypoints can easily hamper performance. One possible extension is to consider the geometrical relationships between keypoints. We leave this task of applying spatiotemporal constraints as future work.

5.7.1 Official run results

This subsection includes the analysis report and DET plot for each event. Each table and plot includes the official runs from performing event detection runs with Eqaution 5.8 and 5.10. The number of the official runs are limited to three for all events. The *1st* refers to the run that generated the best (lowest) NDCR from



(a) *CellToEar* (b) *Embrace*

(c) *PeopleSplitUp* (d) *PersonRuns*

Figure 5.8: For some events, the detection window has too many keypoints, and it can be challenging to the system.

all official runs for particular event. The *2nd* denotes the 2nd best official run. The *3rd* denotes the 3rd best (i.e., worst out of all three) official run. The DET plots also include the lowest NDCR points from the given runs, which are depicted as solid symbols with the corresponding colors (i.e., red, blue and green, respectively). We also include the best NDCR from 2008 TRECVid participants (the black square dots in DET plots). All NDCR points are actual NDCR, meaning they are computed from the actually submitted runs. All TRECVid results are from either [101] or [2]. The table legends are:

- Ref: total number of references in the testing set

- Sys: total number of detection declaration by system
- NCor: number of correct detection
- NFA: number of false alarm
- NMiss: number of miss
- RFA: false alarm rate
- PMiss: probability of miss
- NDCR: NDCR computed by Equation 5.14.

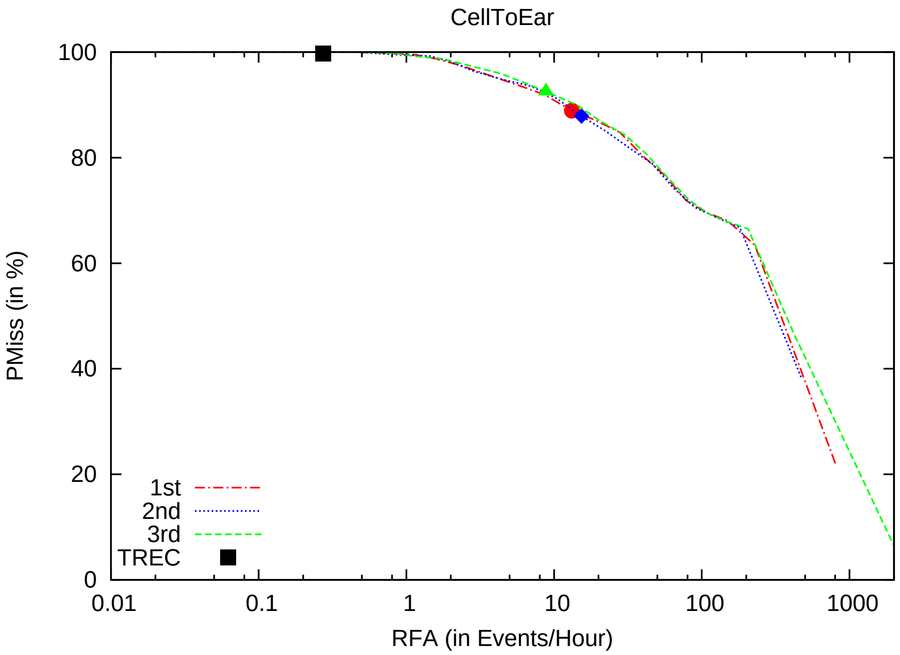


Figure 5.9: DET for *CellToEar*

Table 5.4: *CellToEar* Results

	Ref	Sys	NCor	NFA	NMiss	RFA	PMiss	NDCR
1st	388	724	43	660	345	13.2	0.8891	0.9551
2nd	388	838	47	767	341	15.34	0.8788	0.9555
3rd	388	487	29	441	359	8.82	0.9252	0.9693
TREC	364	15	1	14	363	0.274	0.997	0.999

- 1st: pyramid matching, variation weighting, $\alpha = 0.25$ and $\beta = 0.75$
- 2nd: pyramid matching, variation weighting, $\alpha = 0$ and $\beta = 1.0$
- 3rd: pyramid matching, variation weighting, $\alpha = 0.5$ and $\beta = 0.5$

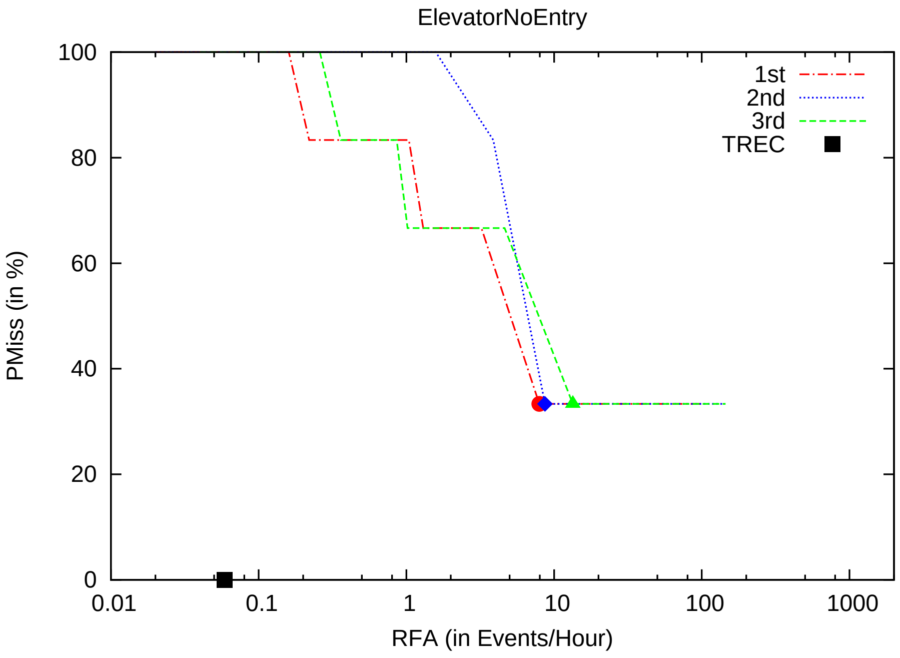


Figure 5.10: DET for *ElevatorNoEntry*

Table 5.5: *ElevatorNoEntry* Results

	Ref	Sys	NCor	NFA	NMiss	RFA	PMiss	NDCR
1st	6	409	4	397	2	7.94	0.3333	0.3730
2nd	6	444	4	434	2	8.68	0.3333	0.3767
3rd	6	685	4	669	2	13.38	0.3333	0.4002
TREC	5	8	5	3	0	0.059	0.000	0.000

- 1st: pyramid matching, variation weighting, $\alpha = 1.0$ and $\beta = 0$
- 2nd: flat matching with $n = 32$, variation weighting, $\alpha = 0.5$ and $\beta = 0.5$
- 3rd: pyramid matching, variation weighting, $\alpha = 0.75$ and $\beta = 0.25$

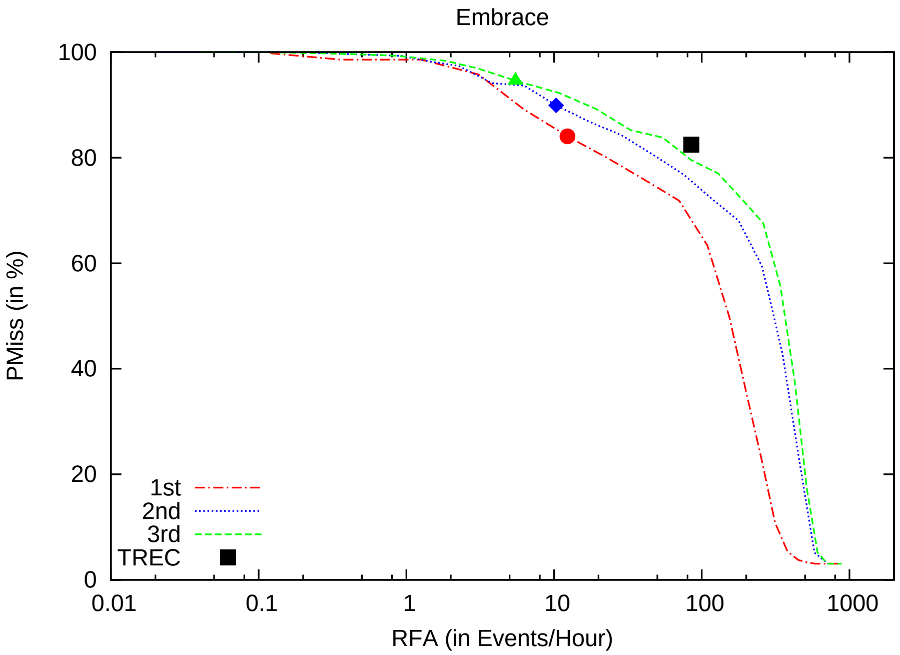


Figure 5.11: DET for *Embrace*

Table 5.6: *Embrace Results*

	Ref	Sys	NCor	NFA	NMiss	RFA	PMiss	NDCR
1st	426	752	68	617	358	12.34	0.8403	0.9021
2nd	426	575	43	517	383	10.34	0.8990	0.9507
3rd	426	298	23	274	403	5.48	0.9460	0.9734
TREC	405	4402	71	4331	334	84.752	0.825	1.248

- 1st: pyramid matching, $\alpha = 0.5$ and $\beta = 0.5$

- 2nd: pyramid matching, variation weighting, $\alpha = 0$ and $\beta = 1.0$

- 3rd: pyramid matching, variation weighting, $\alpha = 0.25$ and $\beta = 0.75$

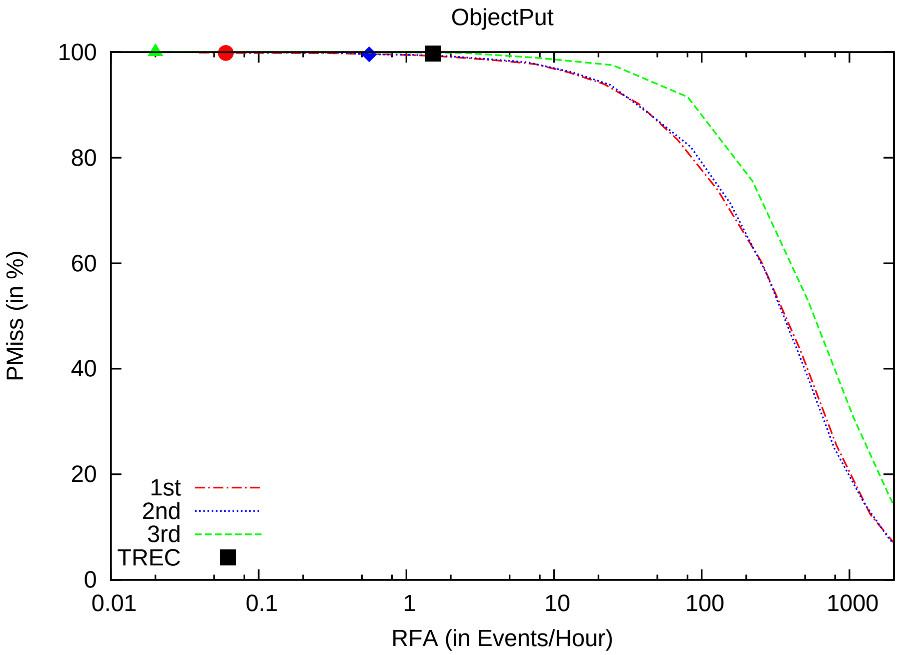


Figure 5.12: DET for *ObjectPut*

Table 5.7: *ObjectPut* Results

	Ref	Sys	NCor	NFA	NMiss	RFA	PMiss	NDCR
1st	2023	6	3	3	2020	0.06	0.9985	0.9988
2nd	2023	38	8	28	2015	0.56	0.9960	0.9988
3rd	2023	1	0	1	2023	0.02	100.0	1.0001
TREC	1958	83	6	77	1952	1.507	0.997	1.004

- 1st: flat matching with $n = 512$, variation weighting, $\alpha = 0.5$ and $\beta = 0.5$
- 2nd: pyramid matching, variation weighting, $\alpha = 0.5$ and $\beta = 0.5$
- 3rd: pyramid matching, $\alpha = 0.5$ and $\beta = 0.5$

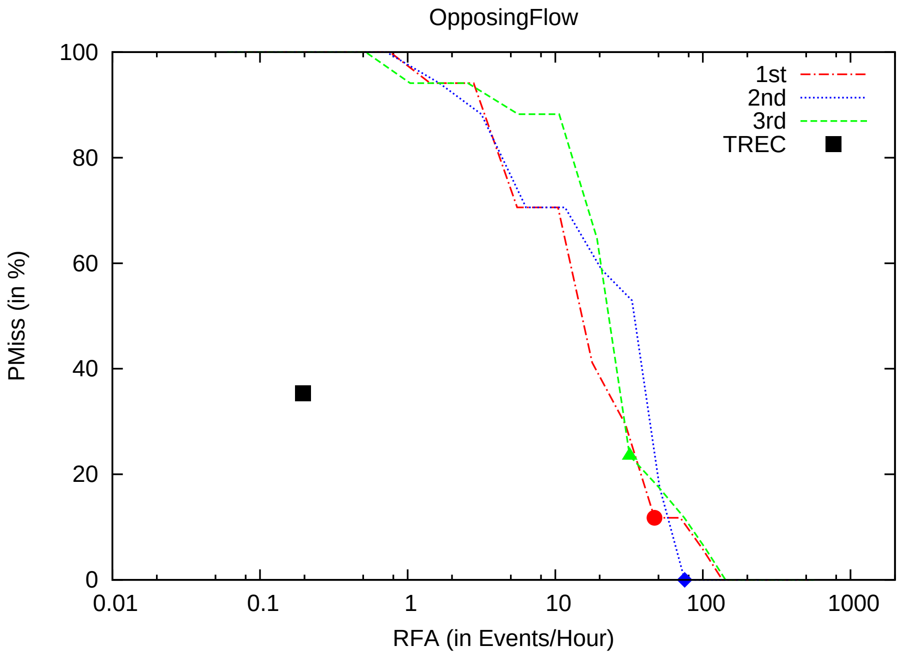


Figure 5.13: DET for *OpposingFlow*

Table 5.8: *OpposingFlow* Results

	Ref	Sys	NCor	NFA	NMiss	RFA	PMiss	NDCR
1st	17	2390	15	2350	2	47.0	0.1176	0.3526
2nd	17	3833	17	3766	0	75.32	0.0	0.3766
3rd	17	1619	13	1597	4	31.94	0.2352	0.3949
TREC	17	21	11	10	6	0.196	0.353	0.354

- 1st: flat matching with $n = 512$, variation weighting, $\alpha = 0.5$ and $\beta = 0.5$

- 2nd: pyramid matching, variation weighting, $\alpha = 1.0$ and $\beta = 0$

- 3rd: pyramid matching, variation weighting, $\alpha = 0.5$ and $\beta = 0.5$

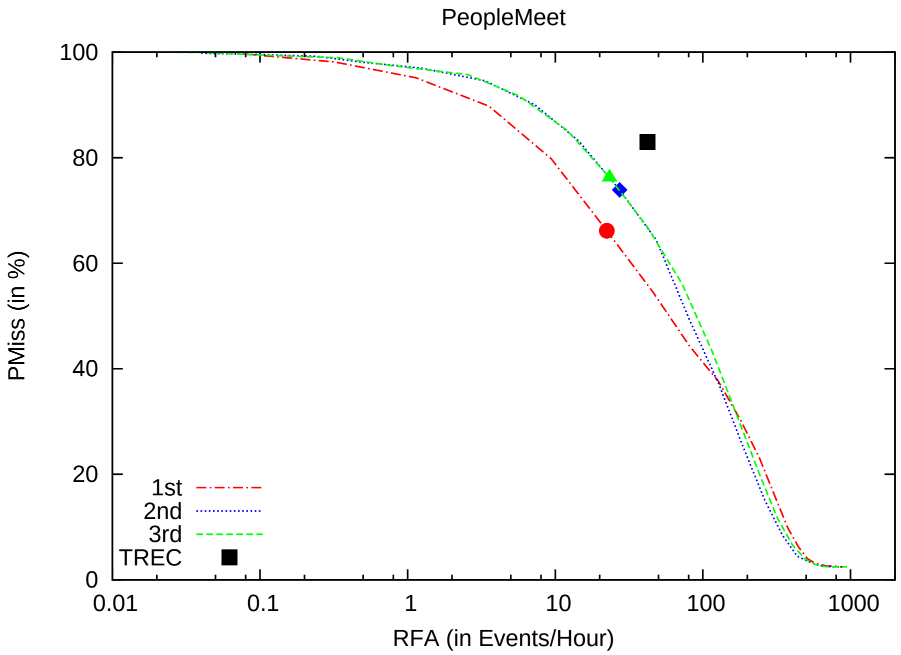


Figure 5.14: DET for *PeopleMeet*

Table 5.9: *PeopleMeet* Results

	Ref	Sys	NCor	NFA	NMiss	RFA	PMiss	NDCR
1st	1291	1808	437	1117	854	22.34	0.6615	0.7732
2nd	1291	1952	337	1366	954	27.32	0.7389	0.8755
3rd	1291	1672	307	1164	984	23.28	0.7621	0.8785
TREC	1249	2578	214	2151	1035	42.09	0.829	1.039

- 1st: pyramid matching, $\alpha = 0.5$ and $\beta = 0.5$
- 2nd: flat matching with $n = 1024$, variation weighting, $\alpha = 0.5$ and $\beta = 0.5$
- 3rd: pyramid matching, $\alpha = 1.0$ and $\beta = 0$

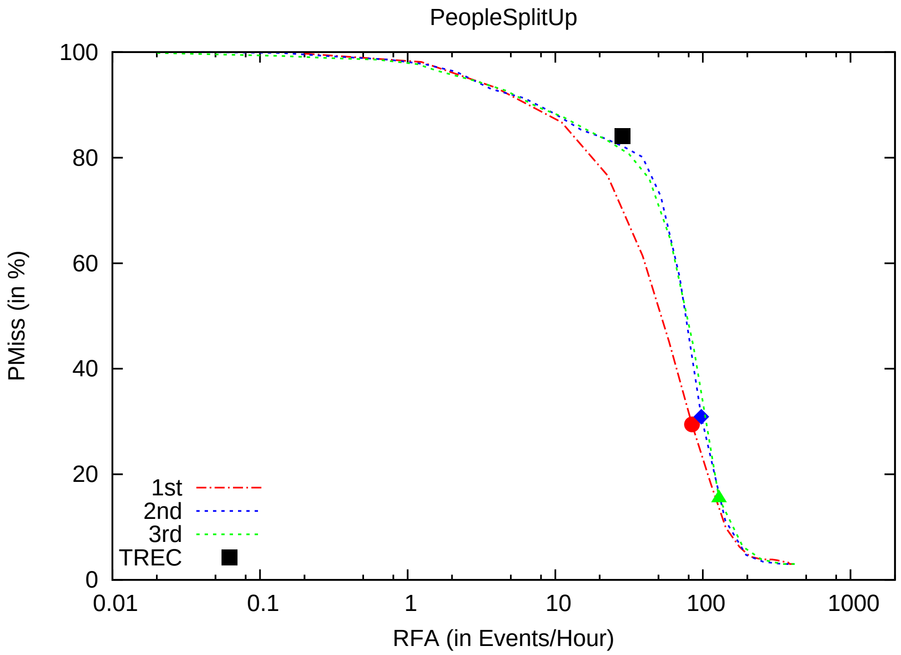


Figure 5.15: DET for *PeopleSplitUp*

Table 5.10: *PeopleSplitUp* Results

	Ref	Sys	NCor	NFA	NMiss	RFA	PMiss	NDCR
1st	699	6038	493	4215	206	84.3	0.2947	0.7162
2nd	699	6859	483	4879	216	97.58	0.3090	0.7969
3rd	699	9108	591	6429	108	128.5	0.1545	0.7974
TREC	681	1721	108	1453	573	28.43	0.841	0.984

- 1st: pyramid matching, $\alpha = 0.5$ and $\beta = 0.5$
- 2nd: pyramid matching, variation weighting, $\alpha = 1.0$ and $\beta = 0$
- 3rd: flat matching with $n = 1024$, variation weighting, $\alpha = 0.5$ and $\beta = 0.5$

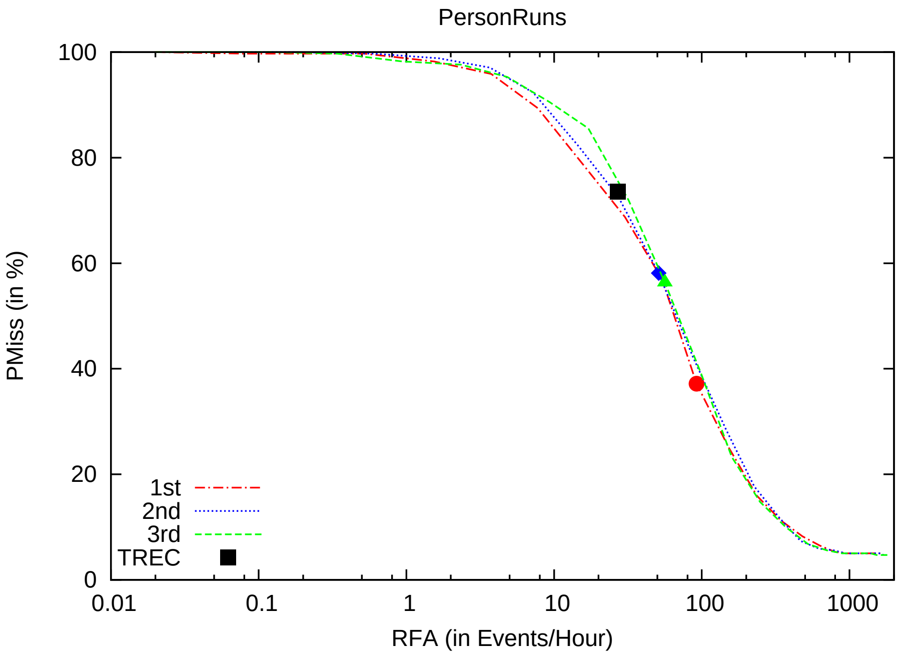


Figure 5.16: DET for *PersonRuns*

Table 5.11: *PersonRuns* Results

	Ref	Sys	NCor	NFA	NMiss	RFA	PMiss	NDCR
1st	339	5025	213	4607	126	92.14	0.3716	0.8323
2nd	339	2800	142	2556	197	51.12	0.5811	0.8367
3rd	339	3076	148	2809	191	56.18	0.5634	0.8443
TREC	321	1463	85	1378	236	26.965	0.735	0.870

- 1st: flat matching with $n = 1024$, variation weighting, $\alpha = 0.5$ and $\beta = 0.5$
- 2nd: pyramid matching, variation weighting, $\alpha = 0.5$ and $\beta = 0.5$
- 3rd: pyramid matching, variation weighting, $\alpha = 0.75$ and $\beta = 0.25$

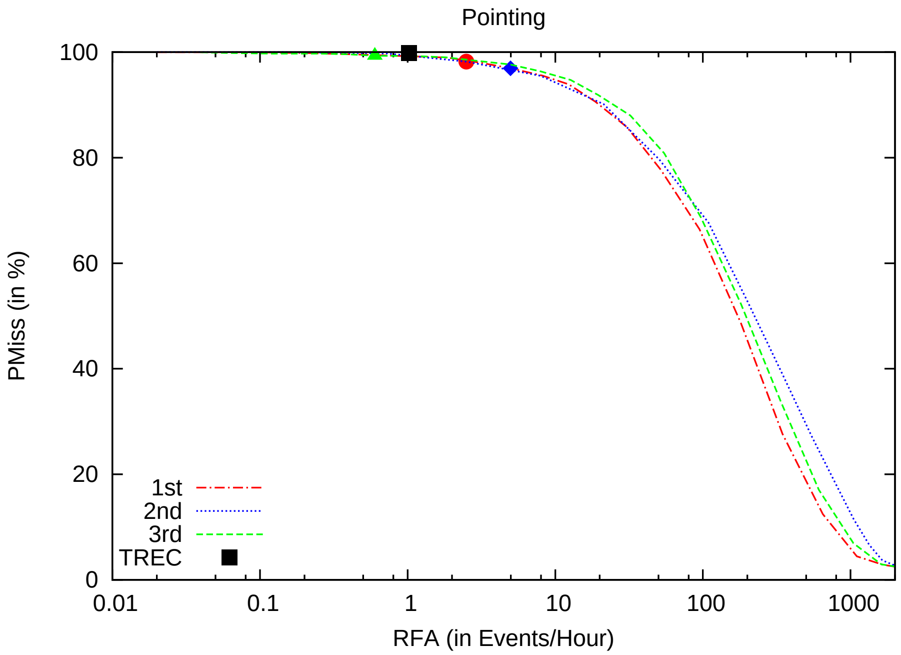


Figure 5.17: DET for *Pointing*

Table 5.12: *Pointing* Results

	Ref	Sys	NCor	NFA	NMiss	RFA	PMiss	NDCR
1st	2426	164	44	125	2382	2.5	0.9818	0.9944
2nd	2426	335	74	249	2352	4.98	0.9694	0.9944
3rd	2426	52	17	30	2409	0.6	0.9929	0.9959
TREC	2369	57	5	52	2364	1.018	0.998	1.003

- 1st: pyramid matching, $\alpha = 0.5$ and $\beta = 0.5$
- 2nd: flat matching with $n = 1024$, variation weighting, $\alpha = 0.5$ and $\beta = 0.5$
- 3rd: pyramid matching, variation weighting, $\alpha = 0.5$ and $\beta = 0.5$

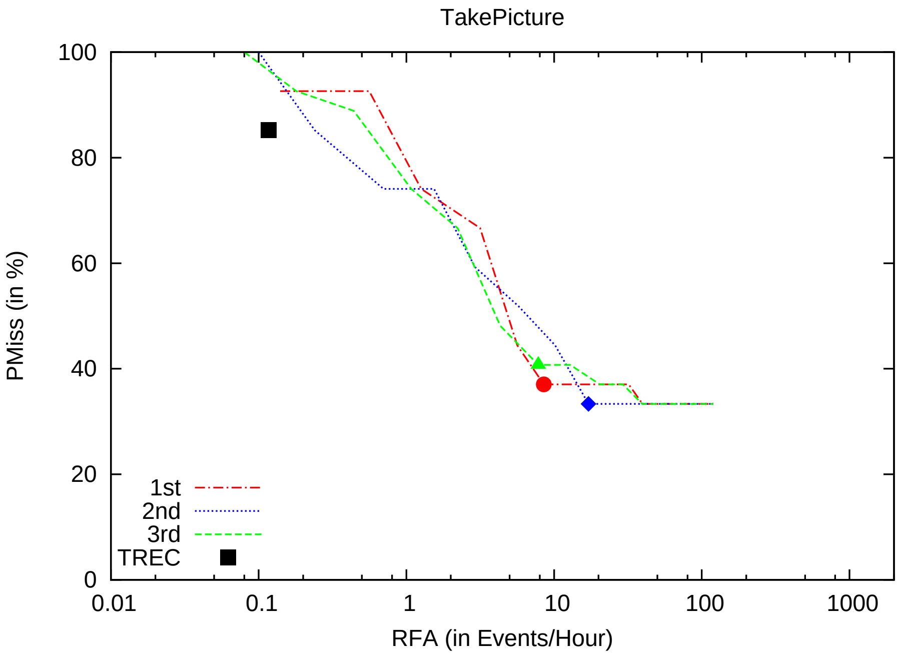


Figure 5.18: DET for *TakePicture*

Table 5.13: *TakePicture* Results

	Ref	Sys	NCor	NFA	NMiss	RFA	PMiss	NDCR
1st	27	459	17	426	10	8.52	0.3703	0.4129
2nd	27	914	18	856	9	17.12	0.3333	0.4189
3rd	27	422	16	391	11	7.82	0.4074	0.4465
TREC	27	10	4	6	23	0.117	0.852	0.852

- 1st: flat matching with $n = 256$, variation weighting, $\alpha = 0.5$ and $\beta = 0.5$
- 2nd: pyramid matching, $\alpha = 0.5$ and $\beta = 0.5$
- 3rd: pyramid matching, variation weighting, $\alpha = 0.5$ and $\beta = 0.5$

CHAPTER 6
TRAIL-BASED VIDEO EVENT RETRIEVAL

6.1 Introduction

Some events can be interpreted as patterns generated by trajectories. Some researchers approach the video event detection problem as matching of temporal trails [18, 21, 71, 47]. Most such attempts view a temporal trail as a spatially flattened trajectory. For instance, a person at location (x_0, y_0) at time t_0, (x_1, y_1) at time t_1 and so on can be marked on a xy-frame and treated as a motion vector. To compare between trajectories, vector similarity can be computed. One problem with this approach is that there is no notion of temporal semantics with the representation. Thus, most work focuses on how to compare the trajectories rather than the application on the actual video data. Perhaps a bigger problem is that most authors are vague about how to obtain the trajectory itself. In this experiment, we discuss our approach on trajectory-based event detection. The trajectory is inherently obtained from a spatiotemporal projection. Based upon our example-based feature matching approach, we provide event examples as both actual and synthetic projection of trajectories. Thus, the goal of this study is not only performing event retrieval but also setting the working foundation for *query-by-example* [35, 117]. We see this as an important step towards building a video event retrieval system on large-scale video data.

6.2 Base feature extraction framework

The choice of features for this task calls for somewhat careful consideration of feature selection. As we have seen in Chapter 5, the local salient features can be effective for wide range of events. However, for the task of event retrieval, the features from the actual example needs to match with the synthetic example. The *strong* features in high dimensional space often capture too much detail that cannot be mimicked by the synthetic examples. We approach this problem by choosing intentionally *weak* features applied on lower scale images. This low-dimensional representation is easier to synthesize and efficient to compare for retrieval purposes.

Color has been one of the most dominant features in vision research. Similarity of color distributions using color moments has been successfully applied in vision retrieval systems [14, 121]. We divide our projection views into non-overlapping subblocks and compare their similarity using the first three moments of each subblock. Since our projection is in 256-bin gray-levels, our color scheme is in a single channel. This approach views an image as a probability distribution of color. Based on probability theory, the probability distribution can be estimated by its method of moments. The choice of color moments over color histograms is made to enhance the retrieval speed with simpler distribution summaries. This feature is usually chosen as a grid descriptor in large-scale retrieval systems [14].

For subblock a of a projection view, the first moment is defined by

$$m_{1,a} = \frac{1}{N} \sum_{j=1}^{N} p_j, \tag{6.1}$$

where p_j is the gray-level value of the j-th pixel and N is the total number of pixels in the subblock a. The second and third moment of the same subblock are then defined as:

$$m_{2,a} = \left(\frac{1}{N} \sum_{i=1}^{N} (p_i - m_{1,a})^2 \right)^{\frac{1}{2}} \tag{6.2}$$

$$m_{3,a} = \left(\frac{1}{N} \sum_{i=1}^{N} (p_i - m_{1,a})^3 \right)^{\frac{1}{3}} . \tag{6.3}$$

Then, the distance d of two subblocks a and b are calculated as absolute differences of the moments:

$$d\left(a,b\right) = |m_{1,a} - m_{1,b}| + |m_{2,a} - m_{2,b}| + |m_{3,a} - m_{3,b}|. \tag{6.4}$$

Thus, the distance of image I at pixel location (x,y) and template T is SAD (Sum of Absolute Differences) at that location:

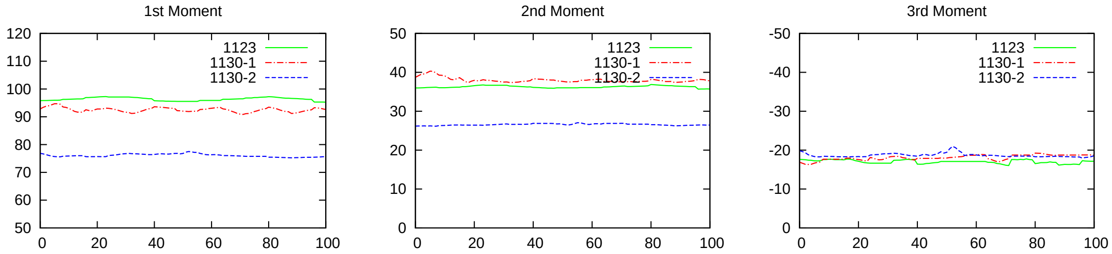
$$D\left(I,T\right) = \sum_{0}^{T_x} \sum_{0}^{T_y} d\left(I_{xy}, T\right), \tag{6.5}$$

where template T has the size of $T_x \times T_y$. Thus, the distance is computed under the area where the template is placed on the projection at (x,y). For temporal detection of event occurrence at t, we compare projection view V and template T using the following distance function. The xt projection view V_{xt} and its template T_{xt} as well as its corresponding ty distance function are the basis:

$$D\left(V_t,T\right) = \alpha D\left(V_{xt}, T_{xt}\right) + \beta D\left(V_{ty}, T_{ty}\right), \tag{6.6}$$

where α and β are user specified weights for each term. Note that the smaller $D\left(V_t,T\right)$ a matching template has, the more similar it is to the input event.

The color moments are susceptible to noise and brightness variation. Considering the encoding artifacts and varying light conditions, this feature is not an optimal



(a) (b) (c)

Figure 6.1: Color moments plot of 3 video segments over 5000 frames

choice. Figure 6.1 shows the actual color moments from three different segments from the same camera location where there is no motion. We can see that the 1st and 2nd moments from *1130-2* are quite far from others even though it was captured in the same day as *1130-1*. Possibly, the real issue here is that there is no way to know what color each event example should have. The same events may have completely different color components even though they have the same pattern. For this, we conjecture that the gradients are better suited to our purpose.

Similar (but much simpler) to [22], we extract oriented gradient points whose magnitude in a given direction exceeds a minimum threshold. The gradient magnitude indicates how quickly the image is changing, while the gradient orientation indicates the direction in which the image is changing most rapidly. Simple 1-D $[-1, 1]$ masks are applied in both x (or y) and t directions. This gives two gradient magnitude components, δx (or δy) and δt, which can be used for orientation:

$$\theta = tan^{-1}\left(\delta t/\delta x\right), \tag{6.7}$$

measured with respect to the x-axis. We convert θ to fit into $0 \leq \theta \leq \pi$. We extract gradient points at eight orientations - i.e., horizontal, vertical and six diagonals. Thus, θ in $\left[0, \frac{\pi}{16}\right)$ and $\left[\frac{15\pi}{16}, \pi\right]$ fall into horizontal where θ in $\left[\frac{7\pi}{16}, \frac{9\pi}{16}\right)$ fall into vertical. We designed this feature to obtain a representation similar to the gist of the image [124, 88, 73].

Both the *xt* and the *ty* projections are inherently vertical component dominant while the horizontal component rarely exists. Thus, we can skip these, and the grid descriptor can be 6-dimensional. As we can see from Figure 6.2, projection with no motion is primarily vertical (bin 5) while some motion introduces more diagonal components (i.e., bin 2, bin 3, bin 4, bin 6, bin 7 and bin 8). Both motion and no motion cases have no gradient elements in bin 1, which is horizontal orientation. Figure 6.3 shows that this feature is somewhat easy to mimic with a synthetic example. This example comes from a *ElevatorOpen* event in *xt* projection. A perfect resemblance may be impossible to obtain but we can see that the synthetic example tending toward the actual example and away from *no motion*.

6.3 Experimental setup

We choose camera 4 data from TRECVid event detection testing set for this task. The data set consists of 5 video segments, each having 2-hour duration. The video data shows two elevator doors with people going in and out of them. The data set original size of 30GB is reduced to about 45MB in the form of projection views, and we based all detection solely on this data. We do not have any training

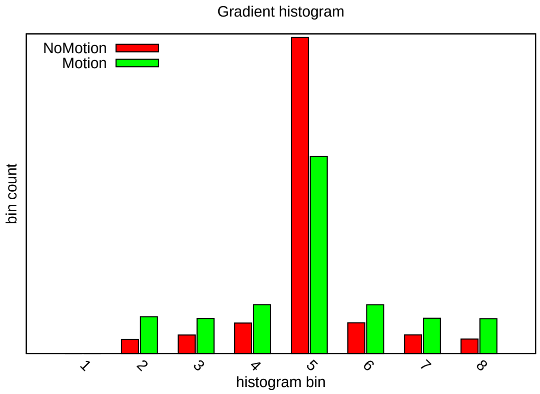


Figure 6.2: Bin-by-bin comparison of gradient histograms from motion and non-motion segments in xt projection

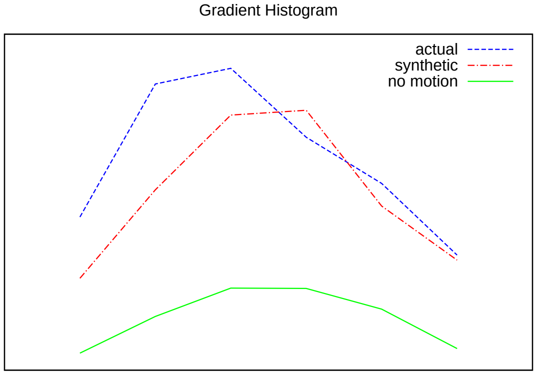


Figure 6.3: Bin-by-bin comparison of gradient gist histogram between actual, synthetic and no motion projection

set for this task since the retrieval is performed with a single user-supplied example. We limit ourselves to only events that can be understood from the motion trail. We selected three events that are prevalent in the data - *elevator2 door open*, *getting into elevator2* and *getting out of elevator2*.

- *PersonIn*: a person going into elevator on the right
- *PersonOut*: a person coming out of elevator on the right
- *ElevatorOpen*: the door on the rightside elevator opens

Table 6.1 shows the occurrence count for each event in the data set.

Table 6.1: Event frequency for trail-based retrieval

Event	Occurrence
PersonIn	21
PersonOut	21
ElevatorOpen	42

We measure performance using standard formulations of precision and recall [26]:

$$P = \frac{Rel \cap Ret}{Ret}, \tag{6.8}$$

and

$$R = \frac{Rel \cap Ret}{Rel}, \tag{6.9}$$

where precision P is the fraction of the retrieved Ret that are relevant Rel, and recall is the fraction of the relevant Rel that are successfully retrieved Ret. For streaming video data, we assume there are no temporal boundaries, and the unit of detection is determined by the example provided by the user. We assume that user-driven consideration of event scale is appropriate for retrieval based on trajectory analysis. Even then, the system needs to determine the temporal stepping size for feature similarity computation. For the event duration of N and the stepping size of n, the temporal span of area under the event consideration will have the minimum $\frac{N}{n}$ overlapping sections. One solution may be achieved by detecting the local maximum similarity. However, the fact that any event occurrence is independent prohibits such a solution. In this thesis, we return any temporal location within the score threshold as Ret. This may result in higher false alram rate penalizing precision. However, we conjecture that more aggresive detection is necessary in this early stage of event detection work. The system performance is compared using the harmonic mean of precision and recall [76]:

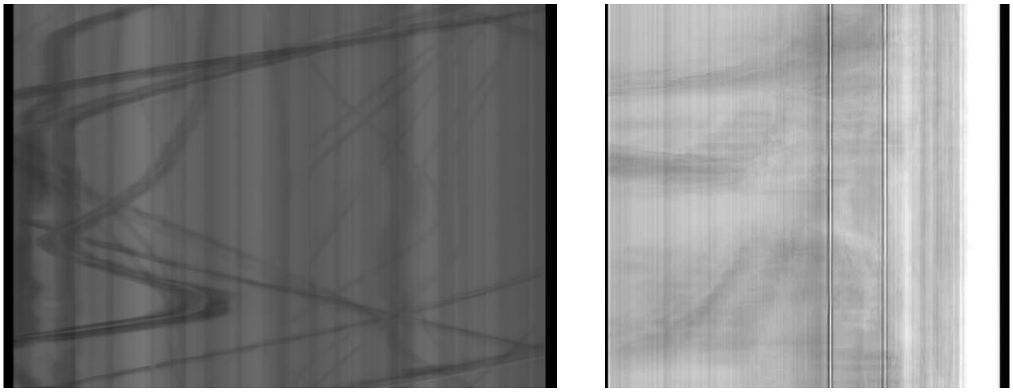
$$F_1 = 2\frac{PR}{P+R}, \tag{6.10}$$

which puts even emphasis on P and R.

6.4 Experimental results

We only performed the retrieval with $\alpha = 1.0$ (and $\beta = 0$). This is because we do not know how to synthetically create the trail information in the ty projection. The results in Chapter 5 indicate that the ty projection contributes significantly in certain event cases. As we can see in Figure 4.10, the artificial video data generates the predictable trail in ty projection. However, we could not find such reliable behavior in real-world data. Figure 6.4 shows *xt* and *ty* examples from our dataset. The *xt* projection clearly shows multiple trajectories discernible even for humans. However, the *ty* projection shows very weak patterns. The strong features in high dimensional space may capture the event salient points in the *ty* projection, but it provides no reliable trajectory information comprehensible to human. This is partly because the main moving figures, a human, are elongated with separately moving limbs, which gives *unclean* projection results. Arguably, the results from Chapter 5 indicate that one should start from $\alpha = 0.5$ and $\beta = 0.5$ for general event detection. Until the synthetic trail generation (and its feature extraction) are studied more, we rely on *xt* projection only for retrieval task. Figure 6.5 shows the example we used for the *ElevatorOpen* event (and we show its gradient histogram in Figure 6.3). This was created with a simple drawing tool available freely. One consideration was given not to use a sharp edge at the end of each trail. This was an attempt to not throw off the gradient histogram one way or another. We set the grid size at 5×5 and only use the projection with 0.5 times the original size. The input video frame is 360×263 with 12.5 frames per second. Thus, the entire projection area is one fourth of the original data.

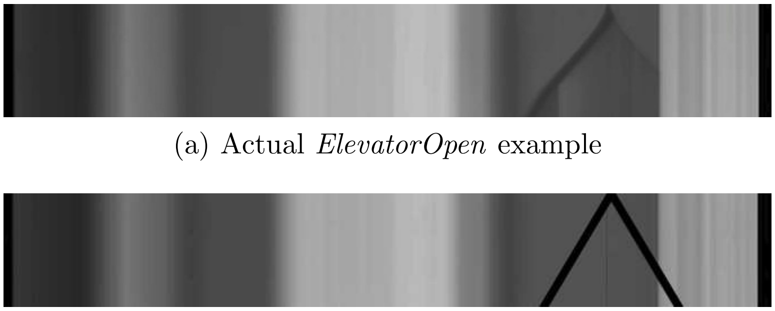
We use the color moments as a baseline and compare its performance with gradient histograms from the actual projection and the synthetic example. The results in Figure 6.6 show that the color moments generally perform the worst, with F1 scores lower than any other settings. While *PersonIn* and *PersonOut* have similar performance, *ElevatorOpen* has higher performance at an F1 score of 0.593. The



(a) xt example

(b) ty example

Figure 6.4: xt and ty projection examples from camera 3. Note that the xt projection shows clear trajectories while it is hard for humans to discern motion information in the ty projection.



(a) Actual *ElevatorOpen* example

(b) Synthetic *ElevatorOpen* example

Figure 6.5: *ElevatorOpen* examples

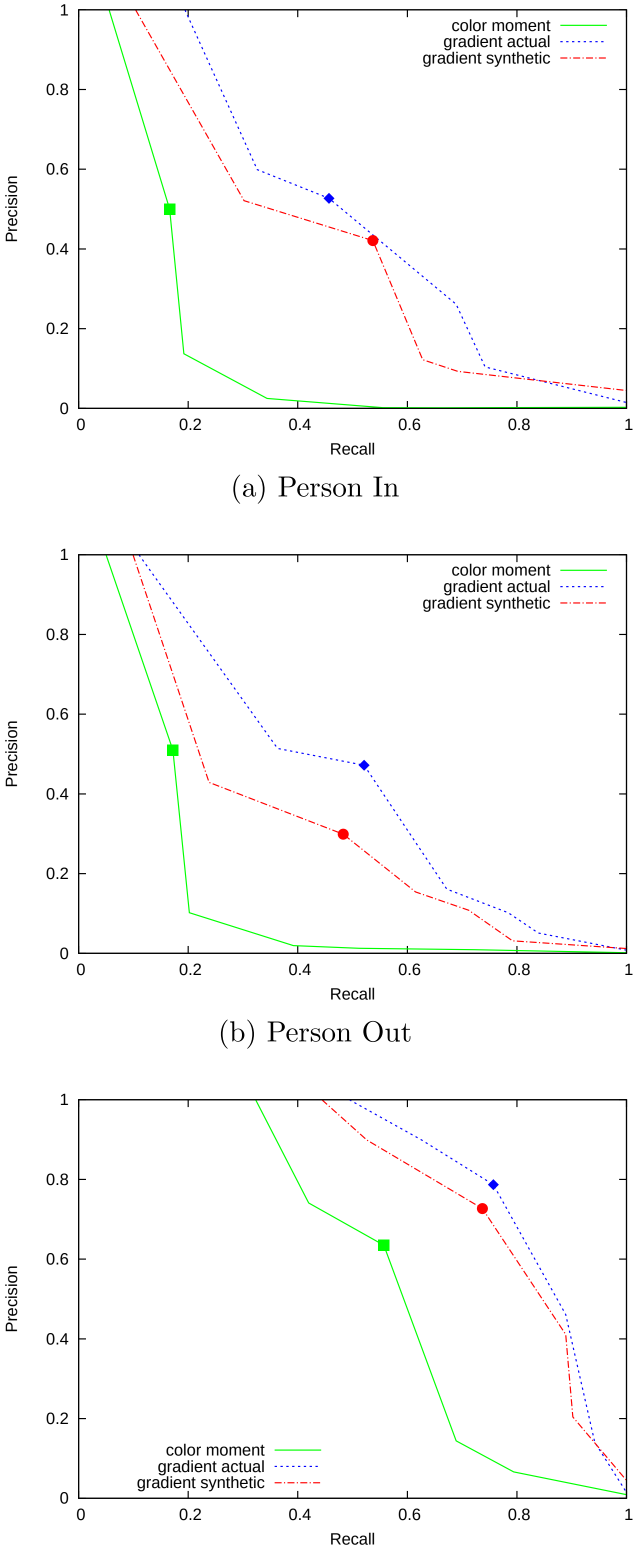
projection given by *ElevatorOpen* is stable and less variable. This results in more similar color and gradient properties within the same event category . As we can see in Figure 6.7, the same *PersonIn* instances have noticable differences in both color and trajectories. This factor generally leads to lower performance.

We also found that the actual projection generally outperform the synthetic projection. This is true across all events we tried. One thing to note here is that the gradients given by the synthetic projection tends to have lower bin counts as we can see in Figure 6.3. This is because the synthetic trajectory is constructed as solid lines while the real trajectories have gradient orientation components within trajectories.

Table 6.2: Event retrieval results (highest F1 score)

	ColorMoment	ActualGradient	SyntheticGradient
PersonIn	0.249	0.489	0.471
PersonOut	0.257	0.495	0.369
ElevatorOpen	0.593	0.771	0.731

Table 6.2 shows the highest F1 score from each event retrieval we tried, and Figure 6.6 the overall performance curves.. The result from *ElevatorOpen* indicates that the stable synthetic example can lead to higher performance. The improved event interpretation process seems critical because even color feature in *ElevatorOpen* shows higher F1 score than either gradient feature in *PersonIn* or *PersonOut*. However, considering the inherent nature of high event variations, achieving the highly stable synthetic example will be challenging future work.

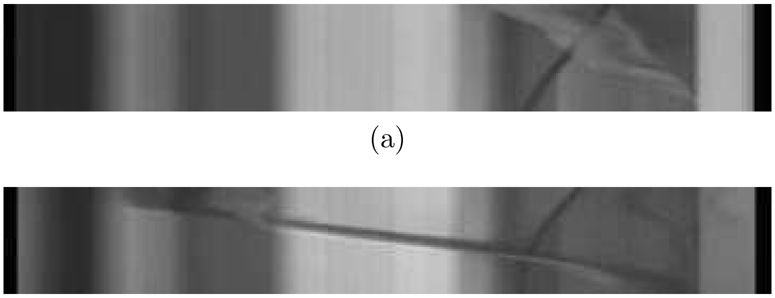


(a) Person In

(b) Person Out

(c) ElevatorOpen

Figure 6.6: Results of each event retrieval



(a)

(b)

Figure 6.7: Examples of two different *PersonIn* instances

CHAPTER 7
CONCLUSION

7.1 Contribution

In this thesis we developed a spatiotemporal projection framework and applied it to event detection and retrieval on large-scale video data. The framework starts by constructing a spatiotemporal 3D volume of streaming video and projecting onto xt and ty domain. We employed the Radon transform as our base projection model. The xt projection captures the horizontal motion pattern within the spatiotemporal volume, and the ty domain captures the vertical pattern. The Radon projection is an inherently lossy summarization of the 3D volume. Motivated by existing 2D vision recognition and classification studies, a sliding window approach to detect a wide range of events was developed. The detector starts by extracting and quantizing local salient points in the projection domain. The events are detected via similarity, computed by the characteristics of codewords defined by the extracted salient points in the 2D projection. The events are defined by a global histogram of codewords, and we investigated various ways to capture their similarities. We also investigated the issue of synthetic event examples to highlight the research direction toward the query-by-example retrieval system on large-scale data.

As enumerated in Chapter 1, a video retrieval system needs to address three fundamental design criteria:

- Scalability: Our projection framework reduces the size of the streaming video considerably. The discussion in Chapter 5 shows that our target video collection of 250GB was reduced to the projection images of 600MB. Data reduction allows a large amount of the temporal video information to be held in memory and processed for event matching. We have shown that detecting a wide range of challenging events is possible even when only considering this intermediate representation.

- Competency: Many systems fail to employ the temporal characteristics of video data, which is crucial for understanding video events. This results in either a highly undescriptive methodology or an application with tightly controlled settings. Projection onto spatiotemporal volumes inherently generates 2D data with temporal extents. 2D feature extraction on the projection domain captures underlying temporal characteristics necessary for content-based event retrieval.
- Robustness: We applied our framework to a wide range of events with a variety of action types such as macro vs. micro or single vs. multiple actors. The results indicate that applying similarity metric is capable of a competitive detection rate to other current research systems. We also showed that our projection framework can be applied to query-by-synthetic-example on large-scale data.

7.2 Future work

Many open research questions as well as exciting avenues for future work exist. We feel that the techniques and methodologies presented in this thesis can be used as basis for more intelligient and efficient video event retrieval system in the future.

7.2.1 Spatiotemporally constrained events

In our current system, we use the global histogram to capture the characteristics of events. Even though we extracted local salient features, we did not consider their geometrical relationships. Any area under the sliding window may have the patterns produced by multiple actors and their interactions. Thus, the final similarity can be easily skewed by irrelevant feature points. This can be a potentially powerful extension to our existing system because the video events tend to have some spatiotemporal constraints.

One possible extension includes a grid-based feature description such as *spatial pyramid matching* [65], where pyramid matching into increasingly fine sub-regions is performed. This can be challenging for streaming data where the exact spatial geometry is hard to obtain but can be readily applied on temporal pyramids. This is because the spatial orientation (i.e., x and y) is not constant in streaming data.

Another idea is to aggregate the existing detection framework with motion path tracking. Thus, instead of having regular subgrid mechanism, the relevant local features that coincide with tracking are given higher detection weights. In our spatiotemporal projection, the motion paths are inherently present and can be readily extracted. Thus, instead of the explicit modelling of individual human path, the lightweight approach can be utilized by considering *weak* features as can be seen in Chapter 6.

7.2.2 Synthetic projection generation

The synthetic event examples, generated based on the principle of projection, open up the possibility of query-by-example of video event retrieval. Video retrieval based on user query has been popular topics in early video retrieval research works. In our approach, the motion path in spatiotemporal volume is projected to *xt* and *ty* image format.

As we have seen in Chapter 6, the synthetic example requires more gradient analysis to achieve the performance of the real projection. More practical approach in this case might be toning down the existing signals. Many vision systems intentionally lower the signal variation by applying the low-pass filter or the Gaussian kernel to reduce the noise. Another popular approaches include a series of morphological operations such as erosion and dilation. A selection of techniques depends on the nature of the data. In our case, the first thing to try could be as simple as taking lower scale projection images, which will effectively produce the narrower motion paths.

From Chapter 5, it is clear that the *ty* projection contains distinct visual cues, but it is somewhat ambiguous when it comes to synthetic trail generation. Even the video segments with low traffic and regular motion generate undistinct projection patterns. One interesting direction is aggregating local salient features and motion tracks, as discussed in Subsection 7.2.1. The insights obtained from this work could point us to better understand what signals are hidden in the *ty* projections. Once we identify the relevant trajectory information in the *ty* projection, the synthetic trajectory can be more effectively utilized.

7.2.3 User interactive system

The development of a video event detection system with synthetic projection allows one to build the user interactive system. User input can be directly interpreted as a 2D projection based on the principle of projection. The notion of spatiotemporal projection significanly reduces the data size. Our experience indicates that the retrieval time is acceptable even for mid-sized data, which was about 10 hours. If we use lower scale projection images, the retrieval time can be improved instantaneously. In this case, the user can expect the retrieval results back in minutes, even with much larger data.

This is especially desirable since touchscreen devices can be used for direct user interaction with a video search system. A mobile device with touchscreen enables the user to search, view and analyze the large-scale video data anywhere and anytime. Not only that, the much-reduced data means that a lot of data processing can be done within the device itself without any server dependence.

One benefit of the user interactive system is a better user experience by interacting directly with what is displayed. In our case, the system can start with a static *xy* frame. Once the system learns more about the data, it can provide with more complicated interactive system. For instance, the system may include the regularly moving objects (e.g., door or elevator) and allows more way of interactions.

REFERENCES

[1] J. K. Aggarwal and Q. Cai. Human motion analysis: A review. *Nonrigid and Articulated Motion Workshop, Proceedings., IEEE*, pages 90–102, 1997.

[2] J. Ajot, J. Fiscus, J. Garofolo, M. Michel, P. Over, T. Rose, and M. Yilmaz. Event detection in airport surveillance. `http://www-nlpir.nist.gov/projects/tvpubs/tv8.slides/event-detection.pdf`, 2008.

[3] C.H. Anderson, J.R. Bergen, P.J. Burt, and J.M. Ogden. Pyramid methods in image processing. *Engineer*, 29(6):33–41, 1984.

[4] E. Ardizzone and M. L. Cascia. Automatic video database indexing and retrieval. *Multimedia Tools and Applications*, 4(1):29–56, 1997.

[5] S. Bakheet, A. Al-Hamadi, B. Michaelis, and U. Sayed. Toward robust action retrieval in video. *Proceedings of the British Machine Vision Conference*, pages 44.1–44.11, 2010.

[6] C. Bartels and G. de Haan. Direct motion estimation in the radon transform domain using match-profile backprojections. *Image Processing, 2007. ICIP 2007. IEEE International Conference on*, pages VI–153–VI–156, 2007.

[7] H Bay, T Tuytelaars, and L Van Gool. SURF: Speeded up robust features. *Computer Vision ECCV 2006*, 3951:404–417, 2006.

[8] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(4):509–522, 2002.

[9] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. *Computer Vision. ICCV 2005. IEEE International Conference on*, 2:1395–1402, 2005.

[10] A. F. Bobick and J. W. Davis. The recognition of human movement using temporal templates. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 23(3):257–267, 2001.

[11] S. Brandt, J. Laaksonen, and E. Oja. Statistical shape features in content-based image retrieval. *International Conference on Pattern Recognition (ICPR)*, 2000.

[12] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems*, 30(1-7):107–117, 1998.

[13] M. Broilo, N. Piotto, G. Boato, N. Conci, and F. De Natale. Object trajectory analysis in video indexing and retrieval applications. *Video Search and Mining*, 287:3–32, 2010.

[14] M. Campbell, A. Haubold, S. Ebadollahi, D. Joshi, M. R. Naphade, A. Natsev, J. Seidl, J. R. Smith, K. Scheinberg, J. Tesic, and L. Xie. IBM research TRECVid-2006 video retrieval system. *Proceedings of the TRECVid 2006*, 2006.

[15] R.J. Campbell and P.J. Flynn. A survey of free-form object representation and recognition techniques. *Computer Vision and Image Understanding*, 81(2):166–210, 2001.

[16] I. Carlbom and J. Paciorek. Planar geometric projections and viewing transformations. *ACM Computing Surveys*, 10(4):465–502, 1978.

[17] M. Celenk, Q. Zhou, and P. Wang. Content-based video indexing and retrieval using the radon transform and pattern matching. *Storage and Retrieval Methods and Applications for Multimedia (SPIE)*, 5307, 2003.

[18] Shih-Fu Chang, W. Chen, H. J. Meng, H. Sundaram, and D. Zhong. A fully automated content-based video search engine supporting spatiotemporal queries. *Circuits and Systems for Video Technology, IEEE Transactions on*, 8(5):602–615, 1998.

[19] E. Chen, Y. Xu, X. Yang, and W. Zhang. Robust event detection scheme for complex scenes in video surveillance. *Optical Engineering*, 50(7), 2011.

[20] L. Chen, M. T. Ozsu, and V. Oria. Robust and fast similarity search for moving object trajectories. *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, pages 491–502, 2005.

[21] S. Dagtas, W. Al-Khatib, A. Ghafoor, and A. Khokhar. Trail-based approach for video data indexing and retrieval. *Multimedia Computing and Systems, 1999. IEEE International Conference on*, 2:235–239, 1999.

[22] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. *Computer Vision and Pattern Recognition. CVPR 2005. IEEE Computer Society Conference on*, 1:886–893, 2005.

[23] M.-S. Dao and N. Babaguchi. A new spatio-temporal method for event detection and personalized retrieval of sports video. *Multimedia Tools and Applications*, 50:227–248, 2010.

[24] P. Daras, D. Zarpalas, D. Tzovaras, and M. G. Strintzis. Shape matching using the 3D radon transform. *3D Data Processing, Visualization and Transmission. 3DPVT 2004. Proceedings. International Symposium on*, pages 953–960, 2004.

[25] R. Datta, D. Joshi, J. Li, and J.Z. Wang. Image retrieval: Ideas, influences, and trends of the new age. *ACM Computing Surveys*, 40(2):5:1–5:60, 2008.

[26] J. Davis and M. Goadrich. The relationship between precision-recall and roc curves. *Proceedings of International conference on Machine learning, ICML 2006*, pages 233–240, 2006.

[27] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. pages 65 – 72, 2005.

[28] Q. Dong, Y. Wu, and Z. Hu. Pointwise motion image (PMI): A novel motion representation and its applications to abnormality detection and behavior recognition. *Circuits and Systems for Video Technology, IEEE Transactions on*, 19(3):407–416, 2009.

[29] S. T. Dumais. Latent semantic analysis. *Annual Review of Information Science and Technology*, 38(1):188–230, 2004.

[30] A. Dyana and S. Das. MST-CSS (Multi-Spectro-Temporal Curvature Scale Space), a Novel spatio-temporal representation for content-based video retrieval. *Circuits and Systems for Video Technology, IEEE Transactions on*, 20(8):1080 –1094, 2010.

[31] A. A. Efros, A. C. Berg, G. Mori, and J. Malik. Recognizing action at a distance. *Computer Vision, 2003. IEEE International Conference on*, 2:726–733, 2003.

[32] J. Evans. The future of video indexing in the BBC. NIST TRECVid Workshop, 2003.

[33] L. Fei-Fei and P. Perona. A bayesian hierarchical model for learning natural scene categories. *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, 2:524–531, 2005.

[34] J. G. Fiscus, J. Ajot, J. S. Garofolo, and G. Doddington. Results of the 2006 spoken term detection evaluation. *SIGIR 2007 Workshop Searching Spontaneous Conversational Speech*, pages 45–50, 2007.

[35] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Qian Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele, and P. Yanker. Query by image and video content: The QBIC system. *Computer*, 28(9):23 –32, 1995.

[36] H. Gao and Z. Yang. Content based video retrieval using spatiotemporal salient objects. *Intelligence Information Processing and Trusted Computing (IPTC), 2010 International Symposium on*, pages 689–692, 2010.

[37] J. Gong, C. H. Caldas, and C. Gordon. Learning and classifying actions of construction workers and equipment using bag-of-video-feature-words and Bayesian network models. *Advanced Engineering Informatics*, In Press, Corrected Proof, 2011.

[38] R. C. Gonzalez and R. E. Woods. *Digital Image Processing (3rd Edition)*. Prentice-Hall, Inc., 2008.

[39] K. Grauman and T. Darrell. The pyramid match kernel: Discriminative classification with sets of image features. *Computer Vision. ICCV 2005. IEEE International Conference on*, 2:1458–1465, 2005.

[40] K Grauman and T Darrell. The pyramid match kernel: Efficient learning with sets of features. *The Journal of Machine Learning Research*, 8:725–760, 2007.

[41] A. Gupta and R. Jain. Visual information retrieval. *Communications of the ACM*, 40(5), 1997.

[42] E. Hadjidemetriou, M. D. Grossberg, and S. K. Nayar. Multiresolution histograms and their use for recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(7):831–847, 2004.

[43] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. Witten. The WEKA data mining software: An update. *SIGKDD Explorations*, 11(1):10–18, 2009.

[44] A. Hanjalic. Shot-boundary detection: Unraveled and resolved? *IEEE Transactions on Circuits and Systems for Video Technology*, 12(2):90–105, 2002.

[45] R. M. Haralick, K. Shanmugam, and I. Dinstein. Textural features for image classification. *IEEE Transactions on System, Man, and Cybernetics*, SMC-3(6):610–621, 1973.

[46] A.G. Hauptmann, M.G. Christel, and Yan R. Video retrieval based on semantic concepts. *Proceedings of the IEEE*, 96(4):602–622, 2008.

[47] J.-W. Hsieh, S.-L. Yu, and Y.-S. Chen. Motion-based video retrieval by trajectory matching. *Circuits and Systems for Video Technology, IEEE Transactions on*, 16(3):396–409, 2006.

[48] Q. Huang, A. Puri, and Z. Liu. Multimedia search and retrieval: New concepts, system implementation, and application. *Circuits and Systems for Video Technology, IEEE Transactions on*, 10(5):679–692, 2000.

[49] C. E. Jacobs, A. Finkelstein, and D. H. Salesin. Fast multiresolution image querying. pages 277–286, 1995.

[50] D. W. Jacobs, D. Weinshall, and Y. Gdalyahu. Classification with nonmetric distances: Image retrieval and class representation. *Pattern Analysis and Machine Intelligence*, 22(6), 2000.

[51] R. Jain. Difference and accumulative difference pictures in dynamic scene analysis. *Image and Vision Computing*, 2(2):99–108, 1984.

[52] R. Jain and H.-H. Nagel. On the analysis of accumulative difference pictures from image sequences of real world scenes. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, PAMI-1(2):206–214, 1979.

[53] F. Jiang, J. Yuan, S. A. Tsaftaris, and A. K. Katsaggelos. Anomalous video event detection using spatiotemporal context. *Computer Vision and Image Understanding*, 115(3):323–333, 2011.

[54] R. Jin and L. Shao. Retrieving human actions using spatio-temporal features and relevance feedback. *Multimedia Interaction and Intelligent User Interfaces*, pages 1–23, 2010.

[55] C.R. Jung, L. Hennemann, and S.R. Musse. Event detection using trajectory clustering and 4-D histograms. *Circuits and Systems for Video Technology, IEEE Transactions on*, 18(11):1565–1575, 2008.

[56] M.B. Kaaniche and F. Bremond. Gesture recognition by learning local motion signatures. *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2745–2752, 2010.

[57] Y. Kawai, M. Takahashi, M. Sano, M. Fujii, M. Shibata, N. Yagi, and N. Babaguchi. NHK STRL at TRECVid 2008: High-level feature extraction and surveillance event detection. *Proceedings of the TRECVid 2008*, 2008.

[58] Y. Ke and R. Sukthankar. PCA-SIFT: A more distinctive representation for local image descriptors. *Computer Vision and Pattern Recognition. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, 2:506–513, 2004.

[59] Y. Ke, R. Sukthankar, and M. Hebert. Efficient visual event detection using volumetric features. *Computer Vision. ICCV 2005. IEEE International Conference on*, 1:166–173, 2005.

[60] Y. Ke, R. Sukthankar, and M. Hebert. Event detection in crowded videos. *Computer Vision. ICCV 2007. IEEE 11th International Conference on*, pages 1 –8, 2007.

[61] J.S. Kim and R.H. Park. Feature-based block matching algorithm using integral projections. *Electronics Letters*, 25(1), 1989.

[62] J.S. Kim and R.H. Park. A fast feature-based block matching algorithm using integral projections. *IEEE Journal on Selected Areas in Communications*, 10(5), 1992.

[63] K.-T. Lai, C.-H. Hsieh, M.-F. Lai, and M.-S. Chen. Human action recognition using key points displacement. *Image and Signal Processing*, 6134:439–447, 2010.

[64] I. Laptev. On space-time interest points. *International Journal of Computer Vision*, 64:107–123, 2005.

[65] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. *Computer Vision and Pattern Recognition. CVPR 2006. IEEE Computer Society Conference on*, pages 2169–2178, 2006.

[66] S. C. Lee, C. Huang, and R. Nevatia. Definition, detection, and evaluation of meeting events in airport surveillance videos. *Proceedings of the TRECVid 2008*, 2008.

[67] M. S. Lew, N. Sebe, C. Djeraba, and R. Jain. Content-based multimedia information retrieval: State of the art and challenges. *Multimedia Computing, Communications, and Applications. ACM Transactions on*, 2(1):1–19, 2006.

[68] J.P. Lewis. Fast normalized cross-correlation. *Vision interface*, 1995.

[69] L. Lijie and F. Guoliang. Combined key-frame extraction and object-based video segmentation. *IEEE Transactions on Circuits and Systems for Video Technology*, 15(7):869– 884, 2005.

[70] T. Lindeberg. Scale-space. *Encyclopedia of Computer Science and Engineering*, 4:2495–2504, 2008.

[71] J. J. Little and Z. Gu. Video retrieval by spatial and temporal structure of trajectories. *Proceeding of SPIE Storage and Retrieval for Media Databases*, 4315(545), 2001.

[72] Y. Liu, D. Zhang, G. Lu, and W.-Y. Ma. A survey of content-based image retrieval with high-level semantics. *Pattern Recognition*, 40(1):262–282, 2007.

[73] D. Lowe. Towards a computational model for object recognition in IT cortex. *Biologically Motivated Computer Vision*, 1811:141–155, 2000.

[74] David G Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60:91–110, 2004.

[75] X. Ma, X. Chen, A. Khokhar, and D. Schonfeld. Motion trajectory-based video retrieval, classification, and summarization. *Video Search and Mining*, 287:53–82, 2010.

[76] J. Makhoul, F. Kubala, R. Schwartz, and R. Weischedel. Performance measures for information extraction. *Proceedings of DARPA Workshop on Broadcast News Understanding*, pages 249–252, 1999.

[77] A. Martin, G. Doddington, T. Kamm, M. Ordowski, and M. Przybocki. The DET curve in assessment of detection task performance. *Proceedings of Eurospeech 1997*, 4:1899–1903, 1997.

[78] A. Martin and M. Przybocki. The NIST 1999 speaker recognition evaluation: An overview. *Digital Signal Processing*, 10:1 – 18, 2000.

[79] A. Mittal and N. Paragios. Motion-based background subtraction using adaptive kernel density estimation. *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, 2:II–302 – II–309, 2004.

[80] T. B. Moeslund, A. Hilton, and V. Kruger. A survey of advances in vision-based human motion capture and analysis. *Computer Vision and Image Understanding*, 104:90–126, 2006.

[81] F. Monay and D. Gatica-Perez. Modeling semantic aspects for cross-media image indexing. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(10):1802–1817, 2007.

[82] M. R. Naphade, J. R. Smith, J. Tesic, S.-F. Chang, W. Hsu, L. Kennedy, A. Hauptmann, and J. Curtis. Large-scale concept ontology for multimedia. *IEEE MultiMedia*, 13(3):86–91, 2006.

[83] C.-W. Ngo, T.-C. Pong, and H.-J. Zhang. Motion analysis and segmentation through spatio-temporal slices processing. *Image Processing, IEEE Transactions on*, 12(3):341–355, 2003.

[84] W. Niblack, R. Barber, W. Equitz, M. Flickner, E. Glasman, D. Petkovic, P. Yanker, C. Faloutsos, and G. Taubin. QBIC project: Querying images by content, using color, texture, and shape. *Storage and Retrieval for Image and Video Databases (SPIE)*, 1908:173–187, 1993.

[85] J. C. Niebles, H. Wang, and L. Fei-Fei. Unsupervised learning of human action categories using spatial-temporal words. *International Journal of Computer Vision*, 79(3):299–318, 2008.

[86] D. Nister and H. Stewenius. Scalable recognition with a vocabulary tree. *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, pages 2161–2168, 2006.

[87] S. A. Niyogi and E. H. Adelson. Analyzing gait with spatiotemporal surfaces. *Motion of Non-Rigid and Articulated Objects. IEEE Workshop on*, pages 64–69, 1994.

[88] A. Oliva and A. Torralba. Building the gist of a scene: The role of global image features in recognition. *Progress in Brain Research*, 155(1):23–36, 2006.

[89] L. Page, S. Brin, R. Motwani, and T. Winograd. The PageRank citation ranking: Bringing order to the Web. *Stanford InfoLab Technical Report*, 1999-66, 1999.

[90] D.-J. Park and D. A. Eichmann. Video event detection as matching of spatiotemporal projection. *Advances in Visual Computing*, 6455:139–150, 2010.

[91] J. Pers, V. Sulic, M. Kristan, M. Perse, K. Polanec, and S. Kovacic. Histograms of optical flow for efficient representation of body motion. *Pattern Recognition Letters*, 31(11):1369–1376, 2010.

[92] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, pages 1–8, 2007.

[93] E. Pogalin, A. W. M. Smeulders, and A. H. C. Thean. Visual quasi-periodicity. *Computer Vision and Pattern Recognition*, 2008.

[94] G. Qian, S. Sural, Y. Gu, and S. Pramanik. Similarity between euclidean and cosine angle distance for nearest neighbor queries. *Proceedings of the 2004 ACM symposium on Applied computing*, pages 1232–1237, 2004.

[95] R.J. Radke, S. Andra, O. Al-Kofahi, and B. Roysam. Image change detection algorithms: A systematic survey. *Image Processing, IEEE Transactions on*, 14(3):294–307, 2005.

[96] K. Rapantzikos, Y. Avrithis, and S. Kollias. Spatiotemporal features for action recognition and salient event detection. *Cognitive Computation*, 3(1):167–184, 2011.

[97] B.S. Reddy and B.N. Chatterji. An FFT-based technique for translation, rotation, and scale-invariant image registration. *Image Processing, IEEE Transactions on*, 5(8):1266–1271, 1996.

[98] W. Ren, S. Singh, M. Singh, and Y.S. Zhu. State-of-the-art on spatio-temporal information-based video retrieval. *Pattern Recognition*, 42(2):267–282, 2009.

[99] Y. Ricquebourg and P. Bouthemy. Real-time tracking of moving persons by exploiting spatio-temporal image slices. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(8):797–808, 2000.

[100] M. Roach, J. Mason, L. Xu, and F. Stentiford. Recent trends in video analysis: A taxonomy of video classification problems. *Proceedings of the International Conference on Internet and Multimedia Systems and Applications, IASTED*, 2002.

[101] T. Rose, J. Fiscus, P. Over, J. Garofolo, and M. Michel. The TRECVid 2008 event detection evaluation. *Applications of Computer Vision (WACV), 2009 Workshop on*, 2009.

[102] Y. Rui, T.S. Huang, and Shih-Fu Chang. Image retrieval: Current techniques, promising directions, and open issues. *Visual Communication and Image Representation*, 10(1):39–62, 1999.

[103] S. Sadek, A. Al-Hamadi, B. Michaelis, and U. Sayed. An efficient method for real-time activity recognition. pages 69–74, 2010.

[104] V. Saligrama, J. Konrad, and P. Jodoin. Video anomaly identification. *Signal Processing Magazine, IEEE*, 27(5):18–33, 2010.

[105] G. Salton. Developments in automatic text retrieval. *Science*, 253:974–980, 1991.

[106] G. Salton and M.J. McGill. *Introduction to Modern Information Retrieval.* McGraw-Hill, Inc., New York, NY, USA, 1986.

[107] S. Santini and R. Jain. Similarity measures. *Pattern Analysis and Machine Intelligence*, 21(9), 1999.

[108] E. Shechtman and M. Irani. Space-time behavior-based correlation-or-how to tell if two underlying motion fields are similar without computing them? *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29:2045–2056, 2007.

[109] S.-O. Shim and T.-S. Choi. Edge color histogram from image retrieval. *International Conference on Image Processing*, pages 957–960, 2002.

[110] T. Sikora. The MPEG-7 visual standard for content description-an overview. *IEEE Transactions on Circuits and Systems for Video Technology*, 11, 2001.

[111] H. Simpson, P. Over, J. Fiscus, and T. Rose. TRECVid 2008 event annotation guidelines version 1.6. `http://www.itl.nist.gov/iad/mig/tests/trecvid/2008/doc/TRECVid08_Guidelines_v1.6.pdf`, 2008.

[112] S. Singh, W. Ren, and M. Singh. A novel approach to spatio-temporal video analysis and retrieval. *Computer Vision/Computer Graphics CollaborationTechniques*, 5496:106–115, 2009.

[113] J. Sivic, F. Schaffalitzky, and A. Zisserman. Object level grouping for video shots. *International Journal of Computer Vision*, 67:189–210, 2006.

[114] J. Sivic and A. Zisserman. Video google: A text retrieval approach to object matching in videos. *Computer Vision, 2003. Proceedings. IEEE International Conference on*, 2:1470–1477, 2003.

[115] A. F. Smeaton. Techniques used and open challenges to the analysis, indexing and retrieval of digital video. *Information Systems*, 32:545–559, 2007.

[116] A. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain. Content-based image retrieval at the end of the early years. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(12):1349–1380, 2000.

[117] J. R. Smith and Shih-Fu Chang. VisualSEEk: A fully automated content-based image query system. *Proceedings of the fourth ACM International Conference on Multimedia*, pages 87–98, 1996.

[118] C. G. M. Snoek and M. Worring. Concept-based video retrieval. *Foundations and Trends in Information Retrieval*, 2:215–322, 2009.

[119] C.G.M. Snoek, B. Huurnink, L. Hollink, M. de Rijke, G. Schreiber, and M. Worring. Adding semantics to detectors for video retrieval. *Multimedia, IEEE Transactions on*, 9(5):975–986, 2007.

[120] A. Stergiou, A. Pnevmatikakis, and L. Polymenakos. Detecting single-actor events in video streams for TRECVid 2008. *Proceedings of the TRECVid 2008*, 2008.

[121] M. A. Stricker and M. Orengo. Similarity of color images. *Proceedings of Storage and Retrieval for Image and Video Databases (SPIE)*, 2420:381–392, 1995.

[122] M. Swain and D. Ballard. Color indexing. *International Journal of Computer Vision*, 7:11–32, 1991.

[123] H. Tamura, S. Mori, and T. Yamawaki. Textural features corresponding to visual perception. *IEEE Transactions on System, Man, and Cybernetics*, SMC-8(6):460–472, 1978.

[124] A. Torralba, K.P. Murphy, W.T. Freeman, and M.A. Rubin. Context-based vision system for place and object recognition. *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, 1:273–280, 2003.

[125] P. Turaga, R. Chellappa, V. S. Subrahmanian, and O. Udrea. Machine recognition of human activities: A survey. *Circuits and Systems for Video Technology, IEEE Transactions on*, 18(11):1473–1488, 2008.

[126] T Tuytelaars and K. Mikolajczyk. Local invariant feature detectors: A survey. 3(3):177–280, 2008.

[127] J. Wang and Z.-J. Xu. Video analysis based on volumetric event detection. *International Journal of Automation and Computing*, 7:365–371, 2010.

[128] C. Watman, D. Austin, N. Barnes, G. Overett, and S. Thomson. Fast sum of absolute differences visual landmark detector. *Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on*, 5:4827–4832, 2004.

[129] X.-Y. Wei, Y.-G. Jiang, and C.-W. Ngo. Concept-driven multi-modality fusion for video search. *Circuits and Systems for Video Technology, IEEE Transactions on*, 21(1):62–73, 2011.

[130] X. Xue, W. Zhang, Y. Guo, H. Lu, Y. Zhang, Z. Sun, Y. Zheng, S. Zhang, H. Liu, Y. Song, J. Zhang, X. He, K. Li, J. Zhou, and Y. Chen. Fudan university at TRECVid 2008. *Proceedings of the TRECVid 2008*, 2008.

[131] X. Yan and Y. Luo. Action recognition via cumulative histogram of multiple features. *Optical Engineering*, 50(1):017–203, 2011.

[132] J. Yang, Y.-G. Jiang, A. Hauptmann, and C.-W. Ngo. Evaluating bag-of-visual-words representations in scene classification. *Proceedings of the International Workshop on Multimedia Information Retrieval*, pages 197–206, 2007.

[133] X. Yang, R. Zhang, Y. Xu, A. Liu, J. Liu, Z. Lu, X. Chen, E. Chen, Q. Yan, Z. Wang, Y. Song, X. Sheng, B. Xiao, Z. Yu, Z. Chu, H. Su, J. Huang, and L. Song. Shanghai jiao tong university participation in high-level feature extraction, automatic search and surveillance event detection at TRECVid 2008. *Proceedings of the TRECVid 2008*, 2008.

[134] P. Yarlagadda, M. Demirkus, K. Garg, and S. Guler. IntuVision event detection system for TRECVid 2008. *Proceedings of the TRECVid 2008*, 2008.

[135] A. Yilmaz and M. Shah. Actions sketch: A novel action representation. *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, 1:984–989, 2005.

[136] A. Yilmaz and M. Shah. A differential geometric approach to representing the human actions. *Computer Vision and Image Understanding*, 109:335–351, 2008.

[137] L. Zelnik-Manor and M. Irani. Event-based analysis of video. *Computer Vision and Pattern Recognition. CVPR 2001. IEEE Computer Society Conference on*, 2, 2001.

[138] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld. Face recognition: A literature survey. *ACM Computing Surveys*, 35:399–458, 2003.

[139] B. Zitova and J. Flusser. Image registration methods: A survey. *Image and Vision Computing*, 21(11):977–1000, 2003.