Panic Detection in Human Crowds using Sparse Coding

by

Abhishek Kumar

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Systems Design Engineering

Waterloo, Ontario, Canada, 2012

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Recently, the surveillance of human activities has drawn a lot of attention from the research community and the camera based surveillance is being tried with the aid of computers. Surveillance is required to detect abnormal or unwanted activities. Such abnormal activities are very infrequent as compared to regular activities. At present, surveillance is done manually, where the job of operators is to watch a set of surveillance video screens to discover an abnormal event. This is expensive and prone to error.

The limitation of these surveillance systems can be effectively removed if an automated anomaly detection system is designed. With powerful computers, computer vision is being seen as a panacea for surveillance. A computer vision aided anomaly detection system will enable the selection of those video frames which contain an anomaly, and only those selected frames will be used for manual verifications.

A panic is a type of anomaly in a human crowd, which appears when a group of people start to move faster than the usual speed. Such situations can arise due to a fearsome activity near a crowd such as fight, robbery, riot, etc. A variety of computer vision based algorithms have been developed to detect panic in human crowds, however, most of the proposed algorithms are computationally expensive and hence too slow to be real-time.

Dictionary learning is a robust tool to model a behaviour in terms of the linear combination of dictionary elements. A few panic detection algorithms have shown high accuracy using the dictionary learning method; however, the dictionary learning approach is computationally expensive. Orthogonal matching pursuit (OMP) is an inexpensive way to model a behaviour using dictionary elements and in this research OMP is used to design a panic detection algorithm. The proposed algorithm has been tested on two datasets and results are found to be comparable to state-of-the-art algorithms.

Acknowledgments

I would like to express my sincerest appreciation to Professor David A. Clausi and Professor Paul Fieguth for their support and guidance in both scholastic and personal matters over the course of my Masters research. I would also like to thank Professor John Zelek and Professor Steven Waslander for reviewing my thesis.

I would like to thank Dr. Amir Hussain Shabani and Professor Alexander Wong for their valuable suggestions in my work and Apurva Narayan for his help during my thesis writing.

Finally, I would like to acknowledge the support of the Natural Sciences and Engineering Research Council (NSERC) of Canada and by GEOIDE (Geomatics for Informed Decisions, a Network of Centers of Excellence) for their financial support of this research.

Dedication

This thesis is dedicated to all my teachers, my parents and my younger brother, without whom it would not have been possible.

Table of Contents

List of Tables

List of Figures

Nomenclature

AROC	Area under ROC curve	31
DE	Dictionary element	20
HMM	Hidden Markov model	28
HOF	Histogram of optical flow	27
OMP	Orthogonal matching pursuit	6
ROC	Receiver operating characteristic	6
SIFT	Sift invariant feature transform	7
TCFP	Triple color flow presentation	16
TN, FN	true negative, false negative	30
TP, FP	True positive, false positive	30
TPR, FPR	True positive rate, false positive rate	31
$\tilde{\mathbf{b}}$	A quantized motion vector	18
$\mathbf{b}$	A vector containing (u, v)	10
$\mathbb{C}$	A set of vectors containing all possible coordinates in an image	15
$\hat{D}$	A dictionary containing selected dictionary elements	20
$\hat{\mathbf{d}}_\mathbf{i}$	The i^{th} dictionary element in a final dictionary $\hat{D}$	20
$\mathbf{d}_\mathbf{i}$	The i^{th} element in a dictionary D	21

D	An initial dictionary	21
$\mathbb{E}$	The combined error function for panic detection	40
E	Total cost function in the optical flow computation	11
E_{data}	Data constantness constraint in the optical flow computation	11
E_{smooth}	Smoothness constraint in the optical flow computation	11
$\check{\mathbf{f}}$	An unit vector	21
$\hat{\mathbf{f}}_{\mathbf{j}}$	The residue of a vector $\mathbf{f}$ after the j^{th} iteration	21
$\mathbf{f_i}$	The i^{th} training vector	19
$\mathbf{f}$	A feature vector	19
F	A training matrix made of N_s training vectors	19
$\underline{\mathbf{h}}_\iota, \hat{\mathbf{h}}_\iota$	An unnormalized histogram and normalized histogram for a frame ι	36
$\mathbf{h} \equiv \mathbf{h}_\iota$	A normalized and noise corrected histogram of motion estimate for a frame ι	36
$h_{\iota,i}$	The i^{th} element of $\mathbf{h}_\iota$	36
I	An image	10
I'	Reconstructed image using optical flow	18
i, j	Indices	19
$\mathbb{L}$	A distance operator to compute distance between two vectors	20
$\mathbb{m}(\mathbb{x})$	The magnitude of motion at a position $\mathbb{x}$	34
N_K	The total number of dictionary elements in a final dictionary $\hat{D}$	20
N_M	The total number of dictionary elements in an initial dictionary D	21
N_p	The total number of moving pixels in a frame.	34
N_s	The total number of training samples	19

υ, φ	Constants used in the spatial localization of a panic.	45
ξ	A constant to stop an iteration	21
ζ	Total number of element in a vector	19

Chapter 1

Introduction

An anomaly [15] is defined as an irregularity in a system. Anomalies are rare incidents and hence they occur far less frequently than normal behaviors. They are unconstrained and the only way to define them is that they are different from regular behaviors. Hence, anomaly detection is done by modeling normal behaviors; An event is declared an anomaly if its characteristic does not comply with the learnt model. The only challenge of an anomaly detection algorithm lies in developing a good model of regular behaviors. The model should have enough tolerance to allow natural variations of normal behaviour but at the same time sensitive enough to detect an anomaly.

With the increasing demand of surveillance of various human activities, an efficient automated surveillance system to detect anomalies has become important. Anomalies in a human crowd can include a variety of cases, as shown in List 1.1 and most of the existing algorithms typically detect a subset of those anomalies. Panic is defined as an anomaly due to a sudden change in the motion behaviour and it is explained in Sec. 1.1. This thesis proposes an orthogonal matching pursuit (OMP) based computationally inexpensive algorithm to detect panic in human crowds.

1.1 What is panic?

An anomaly in human crowds can appear in many different forms and they can represent varying levels of human safety issues. A few examples of anomalies in human crowds have been listed below.

1.1 Anomaly Types

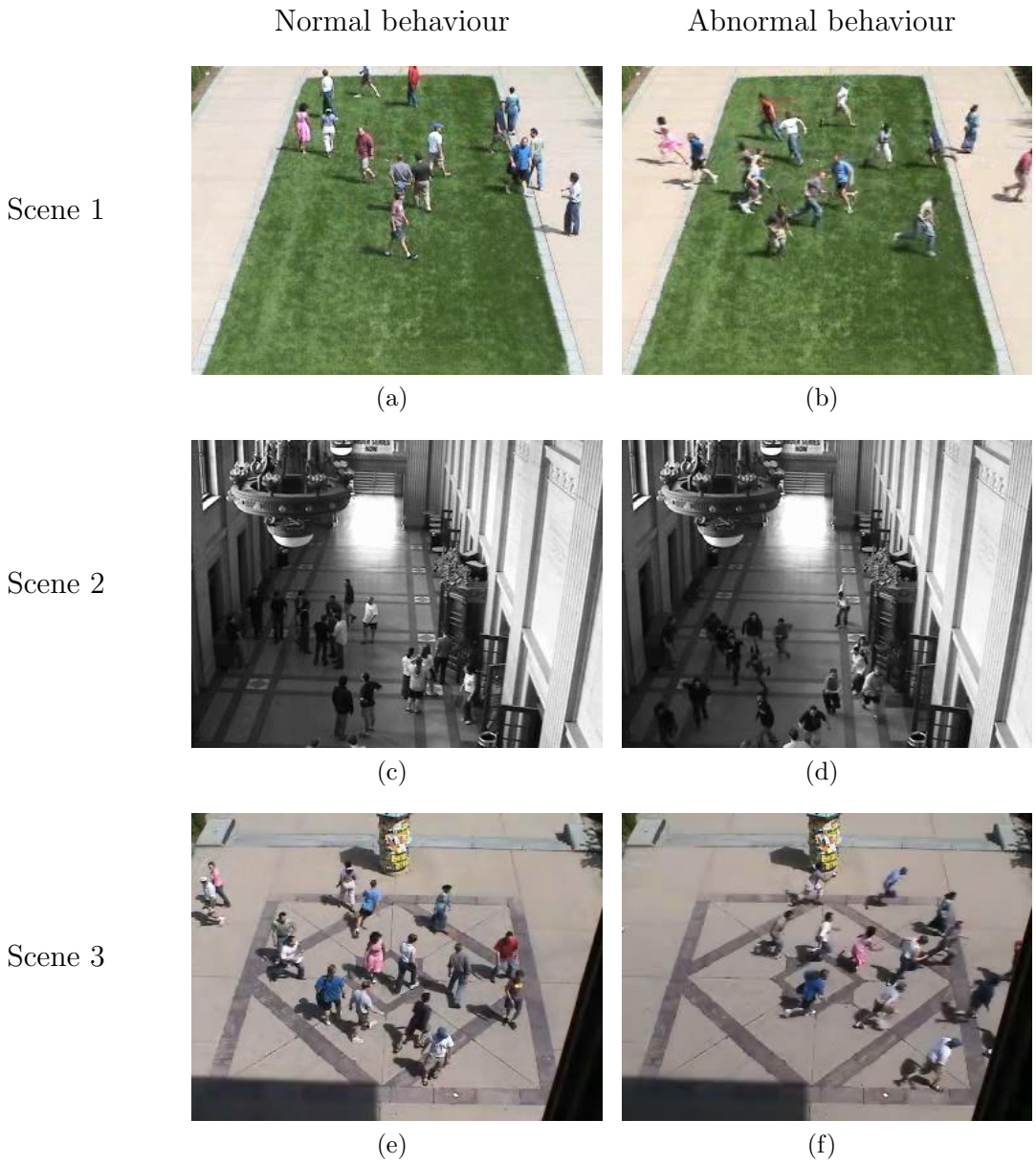


Figure 1.1: Examples of samples from each scene of a dataset from University of Minnesota [1], these scenes are used in Ch. 4. A typical motion behaviour is shown in (a), (c), and (e), where people are walking. Figures (b), (d), and (f), show examples of panic situations when people start to run.

- *In a public area people are walking normally and then suddenly they start to run.*
- *Someone tries to move in the wrong direction in a pedestrian area, a public stairway or escalator.*
- *A vehicle in a pedestrian area.*
- *Animals or pets in an animal prohibited area.*
- *A left behind object in a public place.*
- *A sudden increase in human density in a particular area.*
- *A fight or other such unacceptable activity in a public place.*

Given a variety of events as crowd based anomalies, most algorithms target a specific subset of anomalies. Panic is one such specialization of crowd based anomalies, where a normal behaviour is defined when people are moving with normal speed and panic occurs when an abnormal change in motion behaviour is observed.

Panic Types

- *Everyone in a crowd starts to run.*
- *An individual starts to run.*
- *A group of people start to run.*
- *Sometimes people pause to look at something mysterious or unexpected. Such stopping of a crowd can also be treated as a panic, however such type of behaviour is associated with other contextual information such as duration of pause, location, etc. Such contextual information makes it difficult to detect panic and this thesis will not identify such behaviors.*

1.2 Motivation

A panic event can occur due to a riot, fight, or other such unwanted activities, and all these activities cause losses in terms of lives and wealth. Panic detection in human crowds is important, as it can help us in detecting a panic in real time and thus a whole mishap can be avoided by taking proactive actions. For example, a stampede [19, 22] occurs because

of unacceptably high density of people in a certain region. If it is made possible to timely detect build-ups of high density and proper arrangements are made to diffuse a crowd, a stampede can be avoided with high probability [22].

Often a panic event occurs due to a fight or a chase. It can be useful if those situations are detected automatically and the information is passed to security officers. In public places, sometimes thieves escape with their loot because security officers do not receive alerts promptly. Such incidents are sometime followed by chasing and chaos. An efficient surveillance system will be able to report such irregular activities quickly, providing additional time to security people to catch miscreants.

Surveillance has become easier with the development of inexpensive digital cameras and fast networks. Many cameras have been installed and now almost every important place is under surveillance; however, manual surveillance (Fig. 1.2) does not seem to be an efficient solution. Humans are good at detecting anomalies; however, they are very poor at maintaining attention during mundane tasks. There are a few problems of manual surveillance:

- Manual surveillance implies that a human is watching other human activities and this may be considered as a privacy breach. Such a surveillance system typically has to pass many rules and regulations checks [25].
- Humans are good detectors of an anomaly because of their strong analytical capability and great sense of context; however, their effectiveness drops sharply in unengaging jobs. Anomalies are extremely rare, so operators have to watch normal activities most of the time. It tends to make them bored and inattentive and thus the chances are high of missing anomalies.
- Humans are expensive, they need lots of space and resources; it makes manual surveillance unaffordable for many people in need.

Today's computers are computationally powerful, power efficient, and compact. Being indifferent to mundane jobs, computer vision based panic detection is a natural choice for panic detection. A variety of methods are being invented to automatically detect anomalies in human crowds. With the use of such systems, only those frames will be displayed for manual verifications which are suspected by the anomaly detection algorithm; in the future, human verification might not be needed at all. A simple computer vision technique, which detects motion in a region of interest, can filter out many frames for an anomaly detection in an empty place such as restricted areas and vacant properties. It shows the capacity to which computer vision can help in various surveillance operations.

Figure 1.2: A typical surveillance control room, where an operator is watching at a set of screens for surveillance purposes[1].

1.3 Solution

Panic detection in a human crowd typically requires analysis of motion patterns [62]. An intuitive method of detecting panic in a human crowd is to keep track of each individual in the crowd, use tracking details to learn motion patterns over time, and alarm when there is significant change in the motion characteristics of an individual in that region. Many researchers [46, 36] have proposed panic detection algorithms using this technique. Tracking based systems work in a sparse crowd; however, as a region becomes more crowded, tracking individuals becomes increasingly complex and erroneous [42, 43]. Hence, the use of a tracking based panic detection is limited to only sparsely crowded regions.

To handle panic in a crowded region, a crowd is assumed as a coherent fluid and algorithms have been proposed [5, 6, 43] to model a crowd with the fluid assumption. Results show the effectiveness of the model in detecting panic in human crowds. Though the assumption of coherency in human crowds holds, the degree of coherency varies significantly. For example, the crowd motion is strictly coherent in an escalator but it is irregular in a park. Types of coherency may change model requirements significantly; for example

[1] http://www.theepochtimes.com/news_images/highres/2008-6-9-73932041.jpg

motion direction is an important feature to detect a panic in coherent motion but it may not help much in an incoherent motion.

Most panic detection methods use motion features such as optical flow [63], image gradient [2], or spatio-temporal volume characteristics [32]. These features allow us to analyze dense crowds using characteristics of a crowd rather than individuals. A panic detection algorithm is broadly composed of two segments:

1. *Representing a crowd behaviour in a compact way.* This involves feature extraction and using those features in a compact way such that they show significantly distinct behaviour in the presence of a panic.

2. *Modeling of a normal crowd behaviour to use it for the panic detection.* For a training set, a normal behaviour is learnt using extracted features. A test sample is labeled as a panic if the learnt model shows high difference with the sample.

Once characteristic features are extracted, they are modeled to represent normal motion behaviors. Many approaches [7, 48, 58] use parametric models, however in parametric approach the data characteristics has to be approximated with a standard distribution and in many cases such approximations do not work well. A recently proposed technique, dictionary learning (Sec. 2.3.3) [29] is gaining interest for panic detection [16]. Using a set of training feature vectors, dictionary learning finds a few representative dictionary elements using an optimization process (Eq. 2.16) such that any training data can be reconstructed using those dictionary elements. As training samples correspond to normal behaviors, a panic is signaled if dictionary elements fail to reconstruct a feature vector within an acceptable error limit.

Orthogonal matching pursuit selects a set of optimal dictionary elements and dictionary learning uses OMP or similar approaches in each iteration of optimization. So OMP is relatively inexpensive than dictionary learning. This thesis is proposing a novel algorithm to model normal crowd behaviors using wavelet based orthogonal matching pursuit (OMP) [47]. Like dictionary learning, OMP (Sec. 2.3.3) based panic detection also uses dictionary elements to reconstruct test feature vectors, and a panic is alarmed if the reconstruction error goes beyond an acceptable limit. The proposed algorithm will be tested on publicly available panic samples by University of Minnesota [1] and the Subway dataset [2]. The proposed algorithm has been evaluated by comparing its result with results of three other state-of-the-art methods [16, 43, 61]. The receiver operating characteristic (ROC) [57] and the F1-measure [35] have been used for evaluation purposes.

1.4 Thesis outline

The rest of this thesis is divided into four chapters. Chapter 2 gives a brief summary of related algorithms, various learning mechanisms, and motion estimation techniques. In Chapter 3, a detailed description of various parts of the new approach has been discussed. The chapter explains each step of the algorithm and why that step is needed develop a powerful but simplified technique of panic detection in human crowds. Chapter 4 gives a quantitative analysis of our results and compares them with other state-of-the-art methods. Results are also compared using SIFT flow [38] and optical flow [11] to understand the dependency of the proposed algorithm on the flow type. Finally, Chapter 5 concludes the thesis with a summary on strengths and weaknesses of the proposed algorithm. Possible future projects have also been mentioned which can remove limitations of the proposed method.

Chapter 2

Background on Panic Detection

Chapter 1 introduced how a computer vision assisted anomaly detection algorithm can make a surveillance system efficient and less error prone. An anomaly [15] is defined as a deviation from normal behaviors and it can appear in many forms as explained in Chapter 1 page 1. Each type of anomaly poses a different set of requirements for an anomaly detection system; hence, it is challenging to develop an anomaly detection algorithm to detect all types of anomalies. To simplify this difficult problem, algorithms are typically developed to address a particular subset of anomalies.

In this thesis the discussion is confined to a panic detection algorithm, where a panic is defined at page 3. As panic is a subset of anomaly types, in a few segments of the thesis anomalies are discussed to give a general idea of a panic detection system. This chapter is composed of five sections. Sec. 2.1 gives a broad overview of a typical anomaly detection system. Sec. 2.2 talks about different methods to estimate motion in a video and Sec. 2.3 explains about possible approaches to learn motion behaviors. After a brief explanation of all related topics, Sec. 2.4 briefly discusses how existing algorithms combine necessary elements to produce a panic detection system. Sec. 2.5 explains two methods of accuracy measurement, that will be used in the thesis and finally, Sec. 2.6 concludes the chapter.

2.1 Overview

A human crowd panic detection system is composed of three main segments:

(a) **Crowd behaviour representation**: Any analysis of a crowd's behaviors requires expressing a crowd's characteristics in a tangible form. A set of features is required

that show a significant change in the presence of irregularities. The type of features largely depends on the type of panic to be detected. A few of the features used in detecting a crowd panic are motion information [16], headcount, texture of the image, and the group-size [37, 51].

(b) **Crowd modeling**: Once features are decided to represent crowd behaviors, a model is needed to learn them. Panics are difficult to model because of high variations and rare occurances; so discriminative classifiers such as support vector machine (SVM) [17] are not effective to classify panic and normal behaviors. Panic detection is better suited to hypothesis testing [20, 21], where the hypothesis testing is a formal method to accept or reject a hypothesis. A hypothesis is an assumption about a given instance and hypothesis testing uses two types of hypothesis:

   (a) Null hypothesis: A null hypothesis represents a "no change" situation with respect to the learnt model. In the case of a panic detection system, a normal behaviour acts as a null hypothesis.

   (b) Alternate hypothesis: An alternate hypothesis appears when the null hypothesis is disproved. It represents a change with respect to the model. In a panic detection system, as the panic is defined as a deviation with respect to a normal behaviour, a null hypothesis, it acts as an alternate hypothesis.

   A panic event is detected if a given instance fails to adequately match the learnt model. Like any hypothesis testing, the main challenge of developing a panic detection algorithm is to allow natural variations of a normal behaviour and also to detect anomalies with high accuracy. Broadly, there are three types of modeling methods and Section 2.3 briefly discusses all those types with their advantages and disadvantages.

(c) **Panic localization**: After modeling a normal behaviour, a cost function is formed to test a sample against the model. For a given sample, if the cost function gives higher cost as compared to a threshold Γ, the sample is classified as a panic. There are two types of panic localization:

   (a) Temporal localization [16]: The main objective of a panic detection system is to detect frames which contain panic. With the aid of an accurate temporal localization, only suspected abnormal frames will be shown to operators. It can significantly increase the efficiency and accuracy of a surveillance system.

   (b) Spatial localization [16]: Once a panic is temporally localized, the spatial localization tells the exact location where the system has suspected an abnormal behaviour. Though it is not the primary objective of a panic detection system, it

can greatly improve a surveillance system. It can also help in the quick detection of false positive cases (Fig. 2.9).

2.2 Motion estimation methods

Given two consecutive video frames $I(t)$ and $I(t+\tau)$, where τ is the time difference, a motion estimation gives a motion vector $(u, v)$from a point $I(x, y, t)$ in $I(t)$ to another point in the next image frame $I(t+\tau)$, such that the point $I(x, y, t)$ has moved to the new point $I(x+u, y+v, t+\tau)$. Many methods [11, 38, 59] have been developed to estimate motion, but the central idea of a motion estimate is the same. It assumes that certain properties of a moving point remains unchanged during the motion (constantness property) and motion vectors change smoothly across the neighboring pixels (smoothness constraint). Based on these assumptions, a cost function is formed and the motion is estimated by minimizing the cost function. Motion estimation methods differ in terms of the constantness assumptions and their formulations in the cost function. Broadly, there are two types of motion estimation methods, namely optical flow (Sec. 2.2.1) and SIFT flow (Sec. 2.2.2).

2.2.1 Optical flow

Optical flow was first introduced by Horn and Schunk [27] and their model assumes that the intensity of a pixel remains unchanged during motion. This assumption can be written as

$$I(x, y, t) = I(x+u, y+v, t+\tau) \tag{2.1}$$

The motion components u and v represent the displacement of the pixel $I(x, y, t)$ along x and y directions in an image frame at time $t+\tau$. Henceforth, $\mathbb{x}$ will be used to represent a vector (x, y) and $\mathbf{b}$ will be used to represent a vector (u, v). Based on this convention Eq. 2.1 can be written as

$$I(\mathbb{x}, t) = I(\mathbb{x}+\mathbf{b}, t+\tau) \tag{2.2}$$

Optical flow is estimated by minimizing an energy function [27],

$$E(\mathbf{b}, \tau) = E_{data}(\mathbf{b}, \tau) + \alpha E_{smooth}(\mathbf{b}, \tau) \tag{2.3}$$

where the component E_{data} penalizes changes in the intensity value and the smoothness term E_{smooth} penalizes differences in the optical flow estimate across neighboring pixels.

With the assumption of constant intensity, as suggested by Horn et al. [27], $E_{data}(\mathbf{b}, \tau)$ can be expressed as

$$E_{data}(\mathbf{b}) = \int |I(\mathbb{x} + \mathbf{b}, t + \tau) - I(\mathbb{x}, t)|^2 \, d\mathbb{x} \tag{2.4}$$

Brox *et al.* [11] demonstrated that a better illumination invariance can be achieved by combining the intensity constancy assumption with an assumption of constant intensity gradient. Constant intensity gradient assumes that the intensity gradient of a pixel remains unchanged during a motion. Eq. 2.4 is modified to include gradient constancy as

$$\begin{aligned} E_{data}(\mathbf{b}, \tau) = \int & |I(\mathbb{x} + \mathbf{b}, t + \tau) - I(\mathbb{x}, t)|^2 \\ & + \gamma |\nabla I(\mathbb{x} + \mathbf{b}, t + \tau) - \nabla I(\mathbb{x}, t)|^2 \, d\mathbb{x} \end{aligned} \tag{2.5}$$

where ∇ is a spatial gradient operator and γ is a weight between both the constancy assumptions. While the gradient constancy assumption is less sensitive to illumination variations than the brightness constancy assumption, it is largely violated in scenarios characterized by sudden illumination changes (Fig. 2.1). Many improvements [54, 59] in the optical flow formulations have been done to handle large illumination changes.

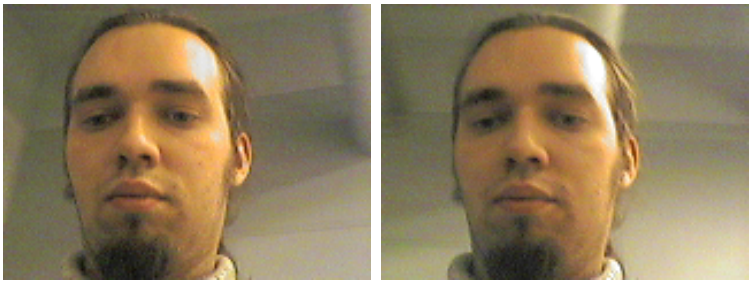
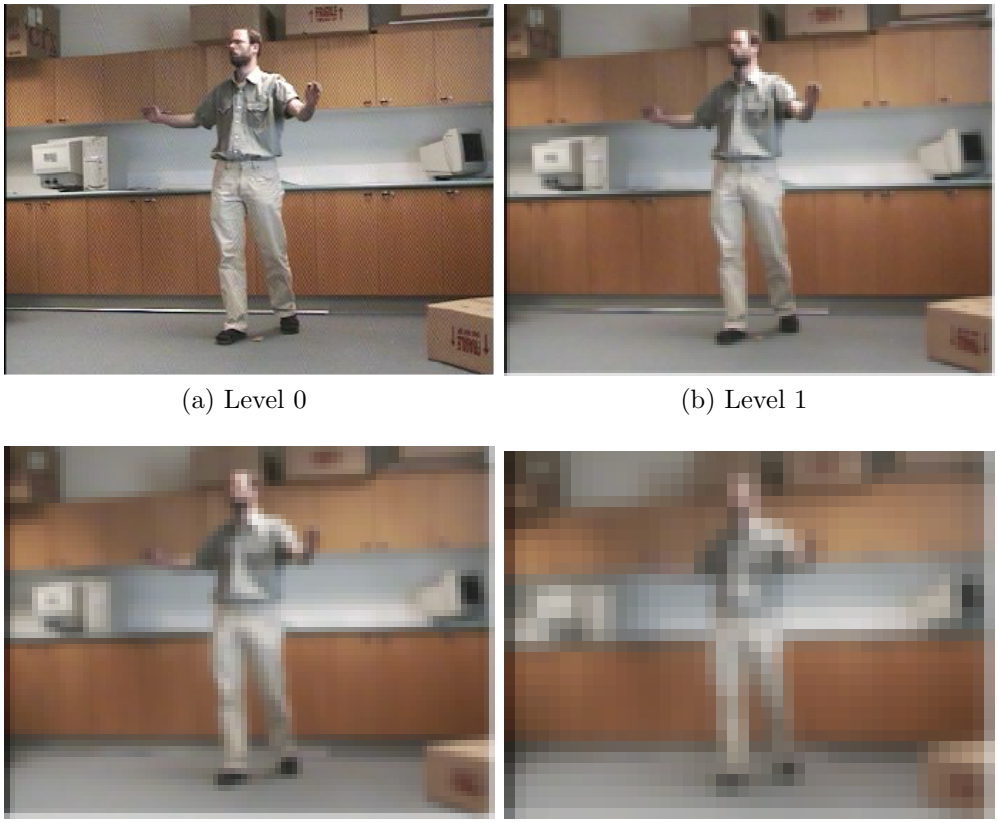


Figure 2.1: An example of strong illumination change, where optical flow tends to give poor results.

With the formulation given in Eq. 2.3, optical flow gives motion estimation for all pixels in a given image; however, it does not work well for large displacements. To solve this problem, Brox *et al.* [11] proposed a warping based optical flow estimation. In the warping based approach, an image pyramid is formed for both image frames $I(t)$ and $I(t + \tau)$. After the warping, a large displacement becomes smaller at the coarser scale, and thus it gives a better estimate of the optical flow at the coarsest scale. The estimated optical flow at the coarser level is used as an initialization for the optical flow estimation



(a) Level 0 (b) Level 1

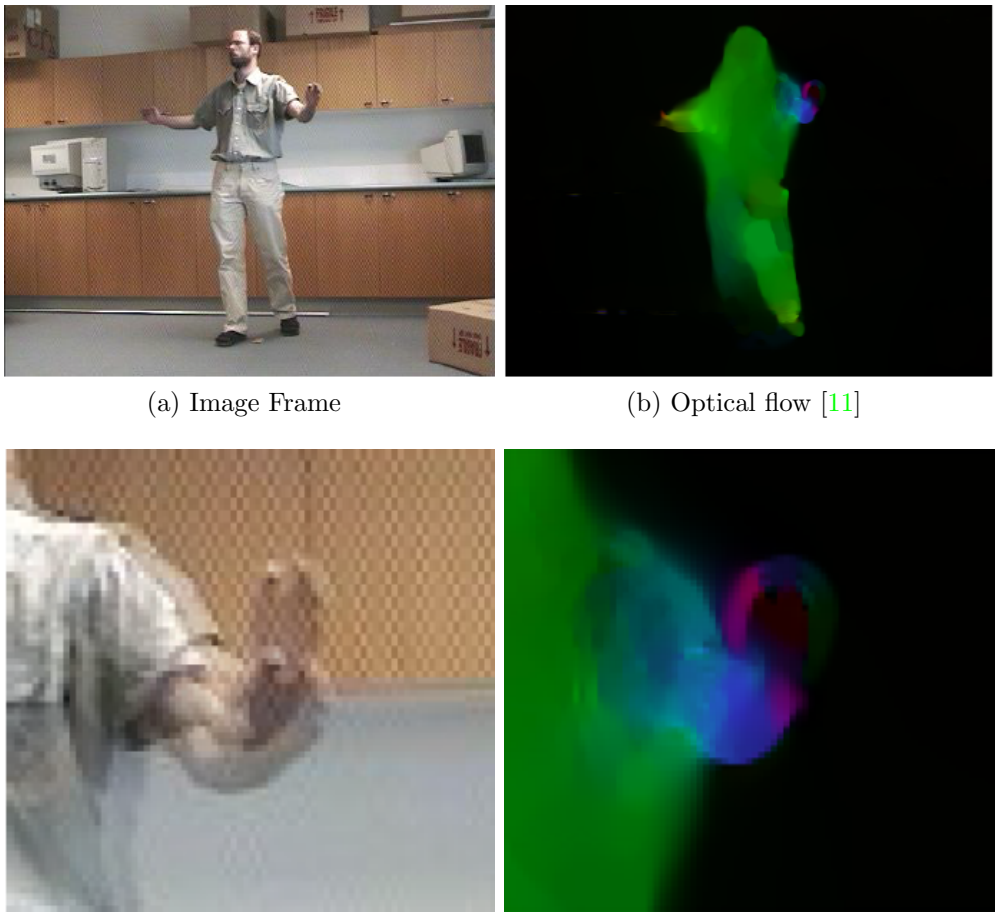
(c) Level 2 (d) Level 3

Figure 2.2: Showing four levels of a Gaussian pyramid [12]. Level 0 corresponds to an original image and level 3 is the image at the coarsest level. For better visualization coarse images are resized to the original size of the image. In the coarsest level (Level 3) the hands are almost indistinguishable.

for the finer level. With good approximation at coarser scale the optical flow estimation improves at finer image. The process continues until the original image is reached.

The main problem with the warping based approach is that as we warp an image to a smaller scale, some information is lost (Fig. 2.2). For a small object, the loss can be so significant that it becomes indistinguishable (Fig. 2.2d). This leads to a high error in the initial estimate and the error continues until the finest scale. Thus warping can introduce a significant amount of optical flow error for a small moving object (Fig. 2.3).

2.2.2 SIFT flow



(a) Image Frame (b) Optical flow [11]

(c) Zoomed in image frame (d) Zoomed in optical flow

Figure 2.3: Despite the fact that the whole palm is moving at the same speed (c), there is a high variation in the optical flow near the palm region (d). The warping based optical flow estimation [11] produces this kind of error due the loss of information at the coarsest level (Fig. 2.2d)[1].

Optical flow gives poor results in the presence of large illumination changes. Also, it gets severely affected by large displacements. SIFT flow [38] has been developed to overcome these limitations of the optical flow. There is no significant difference in the

[1]The image is taken from Brox *et al*''s paper [10].

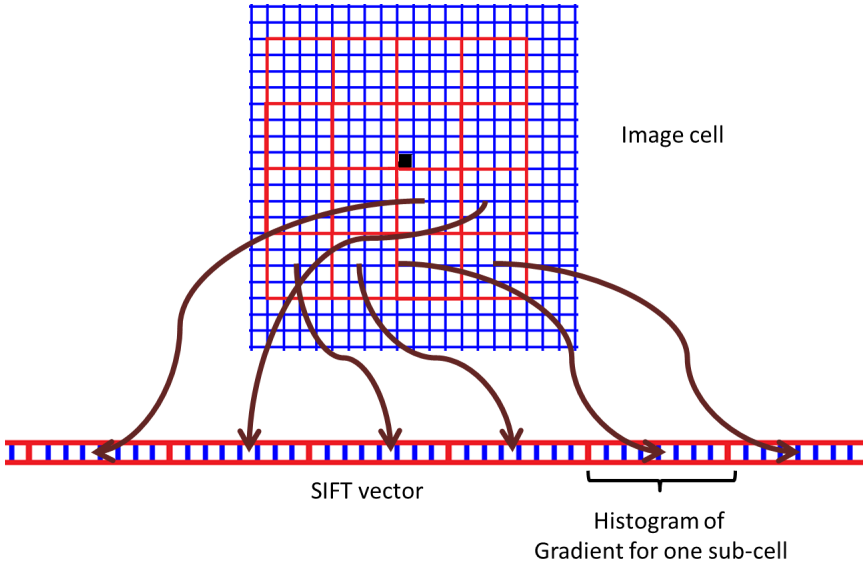


Figure 2.4: Showing a typical cell around a pixel, shown by a black dot, that is used in the SIFT feature formation. The reference gradient direction g is estimated for the pixel by a weighted average of the gradient orientation around the pixel. An eight bin histogram of image gradient is generated for each sub-cell in the cell relative to the g. Histograms of all sub-cells are concatenated to form a SIFT feature vector for the pixel.

formulations of optical flow and SIFT flow, since both make constantness and smoothness assumptions. However SIFT flow uses scale invariant feature transform (SIFT) feature vectors [39] to represent a pixel's property. SIFT uses information from neighboring pixels to provide robustness against illumination variations, scaling and rotations.

In the SIFT flow estimation, SIFT features are estimated for each pixel in both images $I(t)$ and $I(t+\tau)$. Given a patch-size p (typical value 8), a $2p \times 2p$ cell is formed around a pixel and the cell is further divided into 4×4 sub-cells. For each cell, a reference gradient direction g is computed as a weighted average of the image gradient around the center of the cell. Using the reference gradient, an eight bin histogram of gradient orientation is formed for each sub-cell and then all histograms of a cell are concatenated to form a

(a) Frame 1 (b) Frame 2

(c) Optical flow [11] (d) SIFT flow [38]

Figure 2.5: Given two consecutive frames (a) and (b), the optical flow [11] (c) and the SIFT flow [38] (d) are shown in the color coded convention (Sec. 2.2.3). SIFT flow looks pixelated as compared to optical flow. There are a few black regions in (c), these regions correspond to abnormally high flow magnitudes which have been set to zero to filter out erroneous motion estimations. These high values are due to errors in the optical flow estimation. No such error is seen in the SIFT flow estimation (d).

feature vector of 128 elements. SIFT features are used in a cost function as

$$E(\mathbf{b},\tau) = \sum_{\mathbb{x}\in\mathbb{C}} ||\overbrace{\mathbb{S}(\mathbb{x}+\mathbf{b},t+\tau)-\mathbb{S}(\mathbb{x},t)}^{\text{data constantness}}||_1 + \frac{1}{\sigma^2}\sum_{\mathbb{x}\in\mathbb{C}}\Big(\overbrace{u^2(\mathbb{x})+v^2(\mathbb{x})}^{\text{motion constraint}}\Big)+ \sum_{\mathbb{x},\mathbb{x}_1\in\mathbb{C}} \underbrace{min\Big(\alpha|u(\mathbb{x})-u(\mathbb{x}_1)|,d\Big)+min\Big(\alpha|v(\mathbb{x})-v(\mathbb{x}_1)|,d\Big)}_{\text{smoothness constraint}} \tag{2.6}$$

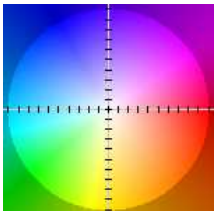
where $\mathbb{x}$ and $\mathbb{x}_1$ are co-ordinates in the co-ordinate space $\mathbb{C}$ of I, and $\mathbb{S}(\mathbf{x})$ is a SIFT feature vector computed for a pixel $I(\mathbf{x})$. The first part of the cost function (Eq. 2.6) is similar to the first part of optical flow Eq. (2.4), with an exception that it uses SIFT feature and $L1$ norm as compared to intensity and $L2$ norm in Eq. 2.4. Some new methods of optical flow [59] have also used $L1$ norm to improve outlier removal. Equation 2.6 also imposes a motion constraint to minimize long motion vectors and a smoothness constraint to limit sudden changes in the estimated motion (u, v) across neighboring pixels. The terms α, d, and σ are constants.

SIFT flow can give accurate motion estimates for large displacements [38]; however it is slower by seven times (Sec. 4.3.4) and it gives somewhat flatter flow estimates (Fig. 2.5d) compared to optical flow (Fig. 2.5c).

2.2.3 Motion representation

Expressing an optical flow in an image is a challenging task. In the initial work by Horn and Schunk [27], optical flow was shown using vectors at each pixel, however, this representation is not effective in showing optical flow for a large image or an image with many details, such as trees, clouds, etc. Sun *at al.* [54] proposed a color coded scheme (Fig. 2.6a, 2.6c) to improve optical flow visualization. In this scheme, the optical flow is expressed in the HSV format. The direction of the motion vector is expressed as hue and the magnitude is presented as saturation of the image. This color coded representation can effectively represent motion for a large image, however, it is not useful for estimating the accuracy of an optical flow. The color coded scheme is useful only if a ground truth of motion estimate is also available in the same representation. Except for a few standard test samples, obtaining ground truths are not possible for real situations; hence, this color coded scheme is not useful to compare the flow results for real cases.

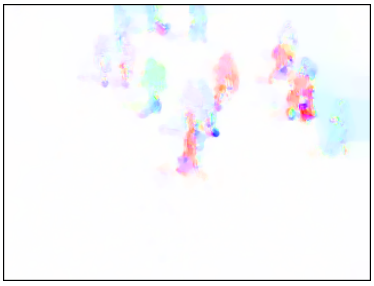
A new technique of optical flow representation, called triple color flow presentation [34] (TCFP) (Fig. 2.6f), has been proposed to remove these limitations. In TCFP, three color channels (red, green, and blue) are used for three different purposes.



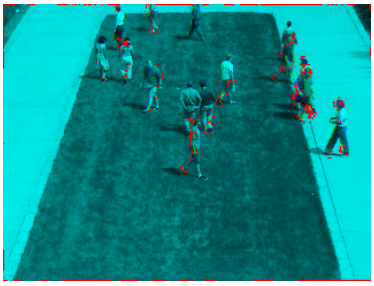
(a)

Image Frames | Color Code | TCFP

(b)

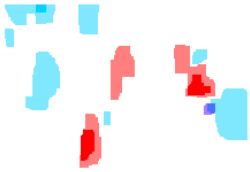


(c) Optical flow

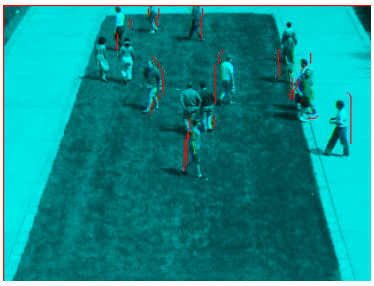


(d) SIFT flow

(e)



(f) Optical flow



(g) SIFT flow

Figure 2.6: (b) and (e) show two consecutive frames of a sample. (a) Color coding used by Sun *et al.* [54]. Color code representation ((c) and (f)) of a motion estimation is useful to see the motion direction but it is difficult to compare the accuracy of optical flow. TCFP ((d) and (g)) provides a mechanism (explained at page 18) of comparing flow results. Red pixels in the case of SIFT flow do not exist on the moving bodies, whereas in optical flow missing pixels are spread over the moving body; however, this difference does not seem to bring a significant difference in the overall accuracy of a motion analysis.

TCFP

Red: *A motion estimate* $\mathbf{b} = (u\ v\ )$ *is unquantized and provides a vector from a pixel in the image* $I(t)$ *to a point in the image* $I(t+\tau)$*. This vector can be used to form the second image* I' *as*

$$I'((\mathbb{x}) + \tilde{\mathbf{b}}, t+\tau) = I(\mathbf{x}, t) \tag{2.7}$$

where motion estimates $\mathbf{b}$ *are quantized in* $\tilde{\mathbf{b}}$*. For an ideal flow estimate,* I' *must be same as* $I(t+\tau)$*. However, a few pixels* $\breve{I}$ *in* I' *do not correspond to a vector from any pixel in* $I(t)$*. This can happen for the following reasons,*

1. *Regions corresponding to* $\breve{I}$ *were occluded or not available in the previous image* $I(t)$*, so no flow was possible.*
2. *An error in the motion estimation.*
3. *A few unmapped pixels may appear because of the conversion of an unquantized motion estimates* $\mathbf{b}$ *to a quantized vector* $\tilde{\mathbf{b}}$*.*

Pixels which are unmapped at frame $t+\tau$ *will have red channel values set to 1, where 1 is the largest value of the color dynamic range. A few red pixels are expected because of aforementioned occlusions and quantization; however a large number of red pixels gives valuable information about the quality of optical flow.*

Green: *The estimated image* I' *is converted to a gray scale image and shown using the green channel.*

Blue: *The original image* $I(t+\tau)$ *is converted to a gray scale image and shown using the blue channel.*

Any existing error in motion estimation can be observed clearly using TCFP. If there are many unmapped pixels then there will be a large number of red pixels. For an idea optical flow the green and blue images should exactly match with each other. In the case of an inaccurate motion estimate, the image in the green channel shall not match with the image in the blue channel, which gives a distinct region in abnormally high blue or green intensity. Thus TCFP provides a framework to analyze and compare the accuracy of a motion estimation algorithm in the absence of the ground truth.

2.3 Modeling methods

Given a set of ζ dimensional training vector $\mathbf{f} \in \mathbb{R}^{\zeta}$, a training matrix F can be formed using N_s vectors as $F = [\mathbf{f_1}, \ , ... \ , \mathbf{f_{N_s}}]$, where $\mathbb{R}^{\zeta}$ is a ζ dimensional vector of real numbers. Broadly, there are three types of models to learn F:

1. Parametric models (Sec. 2.3.1)
2. Non-parametric models (Sec. 2.3.2)
3. OMP and dictionary learning (Sec. 2.3.3)

2.3.1 Parametric models

In a parametric model [20], F is represented in terms of a standard statistical distribution such as a Gaussian distribution [20]. In general, a parametric model makes a few assumptions about the data distribution and extracts a set of N_K parameters $P = \{P_i \quad \forall i \in [1, \ ... \ , N_K]\}$ as

$$P_i = \mathbb{F}_i(F) \quad \forall i \in [1, \ ... \ , N_K] \tag{2.8}$$

where $\mathbb{F}_i$ is a function to extract a parameter P_i from F. The parameter set P represents the training class F. A function $\mathbb{Z}$ computes the probability p that a test sample $\mathbf{f}$ belongs to a training class, represented by the parameters P. $\mathbb{Z}$ is defined based on the parametric model in use.

$$p = \mathbb{Z}(P, \mathbf{f}) \tag{2.9}$$

Usually finding a perfect parametric model for a data set is difficult and an approximate parametric model is selected to represent them. These approximations sometimes causes a high inaccuracy. Finding an appropriate parametric model becomes even more difficult with the increase in the dimension ζ of feature vectors. In a few cases, samples are distributed so that they can not be modeled using a single parametric model and this leads to a mixture of models [7].

2.3.2 Non-parametric models

Unlike a parametric model, a non-parametric model [20] does not assume a particular distribution about samples. It extracts distribution structure from the sample data and uses it to estimate the association of a given sample with the training class. As this approach does not make any assumption about the sample, it is more generic than the parametric model.

One common non-parametric method is the nearest neighbor (NN) [20, 16]. In the NN, a closest training sample $\mathbf{f_{\hat{i}}}$ with respect to a test sample $\mathbf{f}$ is found as

$$\hat{i} = \underset{i \in [1, \ ... \ , N_s]}{\text{argmin}} \ \mathbb{L}(\mathbf{f_i}, \ \bar{\mathbf{f}}) \tag{2.10}$$

where $\mathbb{L}$ is a distance operator to compute distance between two vectors, and $\hat{i}$ is the index of the closest training sample with respect to $\mathbf{f}$.

A constant Γ is used to label a test sample $\mathbf{f}$ as a member of the training class as

$$\begin{cases} \mathbb{L}(\mathbf{f}, \bar{\mathbf{f}}_{\hat{\mathbf{i}}}) \leq \Gamma & \mathbf{f} \text{ belongs to training class} \\ \text{otherwise} & \mathbf{f} \text{ is an outlier} \end{cases}$$

Although the NN method does not make any statistical assumption about the data, it requires a large number of training samples to model a class properly. The requirement of training samples goes higher with an increase of dimensions [30]. Also as the algorithm needs to calculate distance with each training sample, the space requirement of NN methods may be larger than a parametric model.

2.3.3 OMP and dictionary learning

A dictionary learning approach learns a matrix of dictionary elements $\hat{D} = [\hat{\mathbf{d}}_\mathbf{1}, \ ... \ , \hat{\mathbf{d}}_{\mathbf{N_K}}]$ based on the training set F, such that any sample $\mathbf{f}$ belonging to the training class can be reconstructed using elements of $\hat{D}$; where N_K is the total number of dictionary elements and $\mathbf{d_i} \in \mathbb{R}^\zeta$ is the i^{th} dictionary element (DE) with a dimension ζ. These dictionary elements are the representative of training data. Any instance $\mathbf{f}$ can be reconstructed using $\hat{D}$ as

$$\mathbf{f} = \hat{D}\boldsymbol{\eta} \tag{2.11}$$

where $\boldsymbol{\eta}$ is an N_K dimensional coefficient vector of $\hat{D}$. The two approaches to estimate $\hat{D}$ include:

1. Orthogonal matching pursuit (OMP) [47, 41]
2. Dictionary Learning [3, 29]

Orthogonal matching pursuit (OMP)

OMP [47] is an extension of the matching pursuit (MP) algorithm [41]. A sufficiently large dictionary $D = [\mathbf{d_1}, \ \dots \ , \mathbf{d_{N_M}}]$ is initialized. MP provides a greedy way of finding an optimal set of dictionary elements $\hat{D} \subseteq D$ from D. The algorithm runs using multiple iterations and after each iteration, one dictionary element from D is determined. The vector $\hat{\mathbf{f}}_\mathbf{j}$ represents the residue of $\mathbf{f}$ after the j^{th} iteration and $\hat{\mathbf{f}}_\mathbf{0} = \mathbf{f}$. A DE is selected after the j^{th} iteration as

$$\hat{i} = \underset{i \in [1, \ \dots \ , N_M]}{\operatorname{argmax}} |\hat{\mathbf{f}}_{\mathbf{j-1}} \cdot \check{\mathbf{d}}_\mathbf{i}| \tag{2.12}$$

where $\check{\mathbf{d}}_\mathbf{i}$ is a unit vector of the i^{th} DE. The selected $\hat{i}^{th}$ DE $\hat{\mathbf{d}}_{\hat{\mathbf{i}}}$ ensures the maximum contribution [41] of $\hat{D}$ to represent the residue $\hat{\mathbf{f}}_{\mathbf{j-1}}$. $\hat{\mathbf{d}}_{\hat{\mathbf{i}}}$ is included in the dictionary $\hat{D}$ and the coefficient of the selected DE is estimated as

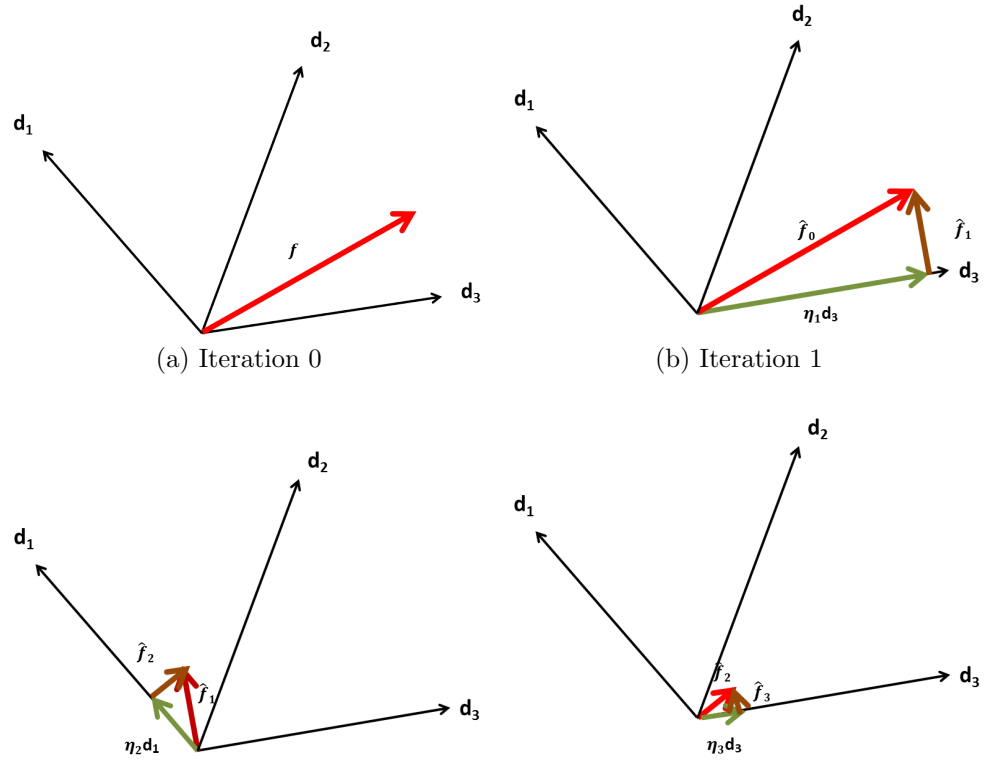
$$\eta_j = \hat{\mathbf{f}}_{\mathbf{j-1}} \cdot \check{\hat{\mathbf{d}}}_{\hat{\mathbf{i}}} \tag{2.13}$$

The estimated coefficient η_j is used in evaluating the residue as

$$\hat{\mathbf{f}}_\mathbf{j} = \hat{\mathbf{f}}_{\mathbf{j-1}} - \eta_j \hat{\mathbf{d}}_{\hat{\mathbf{j}}} \tag{2.14}$$

Fig. 2.7 demonstrates the selection of dictionary elements for a 2-D vector $\mathbf{f}$ using MP, where the initial dictionary D is formed with three 2-dimensional DEs as $D = [\mathbf{d_1}, \mathbf{d_2}, \mathbf{d_3}]$. In the first iteration (Fig. 2.7b), the projection of $\mathbf{f}$ on $\mathbf{d_3}$ is maximum so $\mathbf{d_3}$ is selected. The vector $\eta_1 \mathbf{d_3}$ is a projection vector of $\hat{\mathbf{f}}_\mathbf{0}$ on $\mathbf{d_3}$ and $\hat{\mathbf{f}}_\mathbf{1}$ is the residue after the first iteration. In the second iteration (Fig. 2.7c), $\hat{\mathbf{f}}_\mathbf{1}$ acts as a new vector for the selection of the next dictionary element. MP provides an efficient way of selecting dictionary elements however it does not enforce any constraint on the dictionary element [47]. The unconstrained D allows multiple selections of the same dictionary element. For example, in the third iteration (Fig. 2.7d), an already selected DE, $\mathbf{d_3}$ is selected again because the projection of $\hat{\mathbf{f}}_\mathbf{2}$ is found maximum on $\mathbf{d_3}$. For practical purposes, iterations are stopped when the following condition is met,

$$||\hat{f}_j||_2 < \xi \tag{2.15}$$



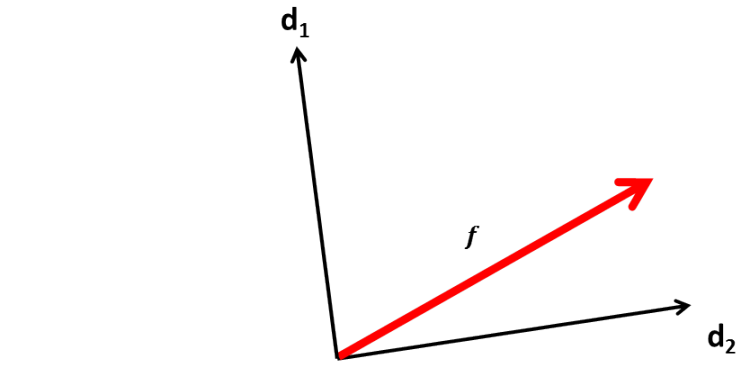
(a) Iteration 0 (b) Iteration 1

(c) Iteration 2 (d) Iteration 3

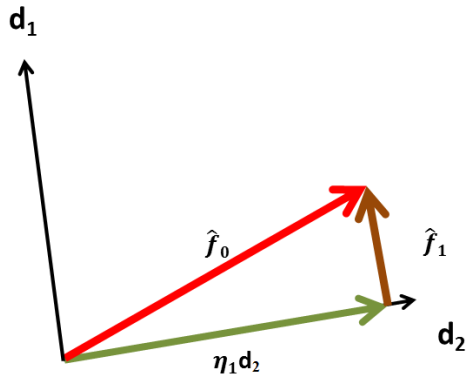
Figure 2.7: Showing three iterations of matching pursuit (MP) [41] for a two dimensional vector $\mathbf{f}$ using a dictionary $D = \{\mathbf{d_1}, \mathbf{d_2}, \mathbf{d_3}\}$ of two dimensional dictionary elements. $\hat{\mathbf{f}}_\mathbf{j}$ is the residue after the j^{th} iteration and $\eta_j \mathbf{d_j}$ is the projection of $\hat{\mathbf{f}}_{\mathbf{j-1}}$ along $\mathbf{d_j}$, where $\mathbf{d_j}$ is the selected dictionary element after the j^{th} iteration. On the third iteration, the residue $\hat{\mathbf{f}}_\mathbf{2}$ selects the same dictionary element that was selected on the 1^{st} iteration, which leaves another residue $\hat{\mathbf{f}}_\mathbf{3}$. MP may continue for many iterations; as, even after three interactions a non-zero residue is left for a 2-D vector and the repeated selection of a DE is possible.

where ξ is a constant to allow the maximum reconstruction residue $\hat{f}$.

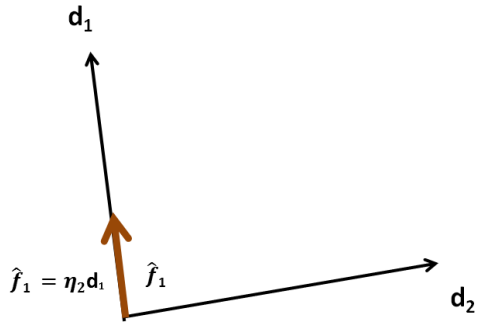
OMP [47] enforces that the residue must be orthogonal to $\hat{D}$. In Fig. 2.7, after the first iteration (Fig. 2.7b) the residue $\hat{\mathbf{f}}_\mathbf{1}$ remains orthogonal to the $\hat{D} = [\mathbf{d_2}]$; however after the second iteration (Fig. 2.7c) the orthogonal condition of OMP is violated as the residue $\hat{\mathbf{f}}_\mathbf{2}$ no more remains orthogonal to all the selected dictionary elements $\hat{D} = \{\mathbf{d_1}, \mathbf{d_2}\}$. To ensure the OMP constraint, $\mathbf{d_1}$ must be orthogonal to $\mathbf{d_2}$.



(a) Iteration 0



(b) Iteration 1



(c) Iteration 2

Figure 2.8: As compared to MP [41] in Fig 2.7, a dictionary $D = \{\mathbf{d_1}, \mathbf{d_2} | \mathbf{d_1} \perp \mathbf{d_2}\}$ of 2-dimensional dictionary elements are used to represent a 2-dimensional vector $\mathbf{f}$. The figure shows two iterations of orthogonal matching pursuit (OMP) [47]. Due to the orthogonality constraint of OMP, the residue $\hat{\mathbf{f}}_\mathbf{1}$ is parallel to the unused dictionary element ($\mathbf{d_1}$) and thus unlike MP (Fig. 2.7), OMP stops at 2^{nd} iteration, which is equal to the dimension of f.

By the orthogonal constraint, OMP enforces that all the selected dictionary elements must be mutually orthogonal. As a ζ dimensional space can only have ζ mutually orthogonal vectors, OMP ensures that to find DEs for a vector f, at most ζ iterations [47] will be required. Fig. 2.8 shows an example of OMP based dictionary selection procedure using an initial dictionary D of two orthogonal dictionary elements, each of 2-dimensions. The figure shows that due to the orthogonality of DEs, it reconstructs a vector $\mathbf{f}$ using D in

two iterations.

For practical applications fewer then ζ dictionary elements are sought, and like MP, iterations are stopped when Eq. 2.15 is satisfied. As OMP requires orthogonal dictionary elements, orthogonal wavelets [47] are a good choice for D to select representative dictionary elements.

Dictionary Learning

Dictionary learning [29], unlike OMP (Sec. 2.3.3), does not use its initial dictionary elements directly in the learnt dictionary and it does not make its initial dictionary from an external source. Given N_s training vectors, an initial dictionary $D \in \mathbb{R}^{\zeta \times N_M}$ is formed using N_M random training vectors, where $N_M \leq N_s$. The initial dictionary D is used in the following optimization function to obtain an optimal dictionary $\hat{D}$ and corresponding optimal coefficients $\boldsymbol{\eta} \in \mathbb{R}^{N_M}$ of N_M dimensions.

$$\min_{\tilde{D}\in\mathbb{C},\tilde{\boldsymbol{\eta}}} \frac{1}{N}\sum_{i=1}^{N}\Big(\underbrace{\frac{1}{2}||\mathbf{f_i} - \tilde{D}\tilde{\boldsymbol{\eta}}||_2^2}_{\text{Reconstruction error}} + \underbrace{\lambda||\tilde{\eta}||_0}_{\text{Sparsity constraint}} \Big) \tag{2.16}$$

where λ is a constant to set the relative weight of sparsity and reconstruction error, $\tilde{D}$ and $\tilde{\boldsymbol{\eta}}$ are the dictionary and the coefficient vector during optimization. At the beginning of optimization $\tilde{D}$ is set to D. $\mathbb{C}$ is a constraint to ensure that the dictionary $\tilde{D}$ contains normalized dictionary elements expressed as

$$\hat{\mathbf{d}}_\mathbf{j}^T \hat{\mathbf{d}}_\mathbf{j} = 1 \quad \forall j = [1, \ ... \ , N_M] \tag{2.17}$$

The optimization function (Eq. 2.16) contains two parts.

1. Reconstruction error: $||\mathbf{f_i} - \tilde{D}\tilde{\boldsymbol{\eta}}||_2^2$ tries to penalize the reconstruction error.
2. Sparsity constraints: $||\tilde{\eta}||_0$ tries to select DEs which can contribute the most to the vector $\mathbf{f}$, thus it imposes that redundant DEs do not get high coefficients. However, solving a L_0 norm is computationally very expensive [29], so the L_1 norm is used instead of L_0 norm as shown below.

$$\min_{\tilde{D}\in\mathbb{C},\tilde{\boldsymbol{\eta}}} \frac{1}{N}\sum_{i=1}^{N}\Big(\underbrace{\frac{1}{2}||\mathbf{f_i} - \tilde{D}\tilde{\boldsymbol{\eta}}||_2^2}_{\text{Reconstruction error}} + \underbrace{\lambda||\tilde{\eta}||_1}_{\text{Sparsity constraint}} \Big) \tag{2.18}$$

The optimization Eq. 2.18 is non-convex in terms of $\tilde{D}$ and $\tilde{\boldsymbol{\eta}}$, however it is convex if one of $\tilde{D}$ and $\tilde{\boldsymbol{\eta}}$ is kept constant. The equation is solved iteratively and in each iteration two steps are followed:

Keep $\tilde{D}$ constant An optimal $\tilde{\boldsymbol{\eta}}$ is estimated by keeping $\tilde{D}$ constant. Optimal $\tilde{\boldsymbol{\eta}}$ can be estimated either by solving Eq. 2.18 or my using a method such as OMP.

Keep $\tilde{\boldsymbol{\eta}}$ constant Optimal $\tilde{D}$ is estimated by solving optimization Eq. 2.18 by treating $\tilde{\boldsymbol{\eta}}$ as a constant.

Iterations stop when Eq. 2.15 is satisfied. The first part of each iteration of dictionary learning uses methods similar to OMP to estimate the optimal $\tilde{\boldsymbol{\eta}}$ and dictionary learning runs for many iterations to converge to the optimal solution, so dictionary learning should be slower than OMP. The difference of time complexity may vary depending on the the number of initial dictionary elements in dictionary learning; the number of initial dictionary elements in the wavelet OMP is kept same as the dimension of feature vectors (Sec. 2.3.3).

As dictionary learning [29] does not impose any constraint on D, it can represent high dimensional samples very effectively. Due to the sparsity constraint, the vector $\tilde{\boldsymbol{\eta}}$ contains high values for only a few DEs. The final coefficient vector $\boldsymbol{\eta}$ of dimension N_K is formed by selecting the N_K largest coefficients from $\tilde{\boldsymbol{\eta}}$ and $\hat{D}$ is formed by including dictionary elements $\tilde{\mathbf{d}}$ corresponding to the selected coefficients. Dictionary learning also allows to modify the cost function to meet various requirements; for example Yang *et al.* [16] modified the cost function in order to enforce smoothness in coefficients across adjacent cells of a video frame. Dictionary learning has been used in panic detection algorithms [16, 63] and shown good results.

2.4 Models for panic detection

Motion characteristics change in most of the cases of anomalies in a human crowd. For example, in the case of panic a group of humans starts to move faster, unexpected motion direction is seen if someone tries to move in the wrong direction, and an object stops to move in a case when someone leaves an unattended object in a public place. Hence, motion plays an important role in panic detection in a human crowd [15]. Depending on the specific type of panic, a system may also use other features too (Sec. 2.1), but motion information remains important. A human crowd panic detection approach can be broadly categorized into two main models based on motion estimation methods, namely object models and particle models.

2.4.1 Object model

An object model considers each individual in the video frame as an object and tries to extract motion information by tracking them. The extracted motion information is used to generate a representative model [14, 18, 28, 31, 45, 56] of the system.

An object model based system usually involves background modeling [49] to extract moving objects. Features such as histogram of oriented gradient (HOG) [17] or scale invariant feature transform (SIFT) [38] are used to detect humans [17] in each frame. Multiple object tracking methods such as multiple-hypothesis tracking [8] and particle filters [9] are used to track detected humans in the crowd. These motion trajectories are clustered together [28] in terms of motion characteristics and spatial positions to learn normal behaviors. Motion characteristics of a test frame are compared with the learnt model and a high deviation from the model implies a higher possibility of a panic.

Basharat *et al.* [7] proposed to track each moving object in the training video and based on those tracks the probability density function (PDF) for each pixel is estimated. The PDF is modeled as a multivariate Gaussian mixture model [7] of motion and the size of the moving object at the corresponding location. During testing, the motion and the size of an object at a pixel form a feature vector at that pixel. The feature vector is compared with the PDF of the a corresponding pixel to compute the probability of normal behaviour. A panic is triggered if abnormal behaviors are detected for a sufficiently large number of pixels. The paper also provides a mechanism to update PDFs in real time.

Michael *et al.* [48] proposed a method of head counting. It used the motion characteristic of head with respect to the torso and the shape of the head. A support vector machine (SVM) [17] is used to detect heads using the histogram of oriented gradient (HOG) [17] feature. This SVM is used for each pixel to get a probability distribution of heads across the whole frame. To reduce false positives it also used a different probability map using optical flow, which gives high probability to those detected heads which move with the same speed as torsos near to the head. The paper gives a headcount in each frame which can be useful in detecting a panic in a crowded region based on the human density.

These object model based methods give high accuracies in a sparse crowd, but in a dense crowd it becomes difficult to keep track of an individual or to detect an object due to severe occlusion. Moreover, the increase in human density increases the complexity of tracking algorithms and reduces the accuracy of tracking. Usually panic detection does not need to know about a specific person but it looks for an overall behaviour of the crowd. These limitations of object models and the idea of using general characteristics of a crowd motivated researchers to look for an alternative model.

2.4.2 Particle model

A particle model considers a crowd as a dense fluid or closely packed particles [5, 16, 32, 43, 56, 61]. It does not need to identify or track an individual person or object in the crowd and thus it avoids problems due to occlusion in a dense crowd. Unlike an object model, particle models use motion information (Sec. 2.2) using differences of consecutive frames in a video. The extracted motion information is broadly called "flow" and usually one of the two types of flow are used: optical flow [11] or SIFT flow [38]. The estimated motion vectors are used to learn the motion behaviour of the crowd and if a test sample shows high error with respect to the learnt model, a panic situation is alarmed.

Shu *et al.* [58] divide each video frame into cells by a fixed grid. Cells are further divided into smaller 4×4 sub-cells. A histogram of motion is estimated for each cell using the average flow vector of all 16 sub-cells. The histogram of motion acts as a feature vector of the corresponding cell and these histograms are clustered using training samples to represent the normal behaviour for that cell. The paper also proposes to use historical information to obtain better detection. It creates a spatio-temporal model for each cell using its neighboring cell in space and time. A correlation function is developed to keep track of incidences of similar histograms between neighboring cells. The correlation function is used to ensure correlated detection of anomalies across cells in space and time.

Mahadevan *et al.* [40] suggested that often used statistical models, such as Markov Random Fields (MRFs) and Latent Dirichlet Allocation (LDA), may help in modeling a crowd behaviour; they fail to keep the visual presentation of the scene, and hence they do not detect spatial anomalies such as detecting abnormal objects (e.g. car, truck) in a pedestrian area. With those models one needs to include other spatial features such as size of object explicitly in the model to detect visual anomalies. They proposed to use dynamic textures to give a joint model of appearance and dynamics. A mixture of dynamic textures (MDT) [13] is used to model temporal normal behaviour and spatial normalcy is modeled using a discriminant saliency detector [23] based on MDT. Temporal normal behaviors helped in localizing anomalous frame and spatial normalcy is used to spatially localize anomalies in a frame. For anomaly detection, test samples are compared with the learnt model and an instance with a low probability is detected as an anomaly.

Although the method proposed by Mahadevan served the purpose of detecting spatial and temporal anomalies, it is slow (2-4 frames per minute) [40]. To solve this problem in a computationally more efficient way, Vikas *et al.* [51] proposed a cell based modeling of motion behaviour. They divided video frames into disjoint cells and defined a feature vector for each cell to contain motion information, texture information, and object size. Background modeling is used to extract foreground for all these processing. Kernel density

functions (KDE) are estimated for each feature parameter independently. The KDE is used to tell whether or not a cell has normal behaviour. The proposed method has been shown to work well, however, background modeling becomes erroneous in a crowded scene and the size of an object has strong dependence on the background model. So, in the absence of good foreground extraction, this system may not give good results.

Previous papers [40, 51] model motion behaviour for each frame but they do not learn motion transition behaviour in time. Learning temporal behaviour of motion can be useful in modeling acceptable motion variations. To model motion characteristics in time and space, Kratz *et al.* [32] divided video into spatio-temporal cuboids and the spatio-temporal gradient is used for each cuboid to obtain motion behaviors. Each cuboid is represented by mean and variance of motion and Kullback-Leibler divergence [33] is used to discriminate motion patterns. A Hidden Markov model (HMM) is used to model local motion pattern and another HMM is used to constrain the transition of motion pattern from one cuboid to its neighboring cuboids. During a panic, the joint probability of spatio-temporal motion transition becomes low and this helps in detecting abnormal behaviors. The main problem of cuboid-based approach is that as these cuboids are arbitrarily placed, important information might get lost across these cuboids leading to inaccuracy in panic detection.

In human crowds, each individual moves with a set of motives and a certain relative distance is also maintained with a neighboring person. This idea led to a social force model [26], which was initially introduced to simulate pedestrian dynamics. It tries to predict human's dynamics in a crowd in the presence of factors such as motivations, opposing forces, group interest, etc. Mehran et al. [43] used the social force model to model crowd motion behaviors. As the social force model can closely emulate the crowd dynamics, Mehran *et al.* tried to interpret anomalies in human crowds by detecting the presence of abnormal social forces in a crowded region using a bag-of-words approach [4].

The proposed approach places a set of particles at a few fixed locations in a video frame and those particles are allowed to advect with the estimated optical flow. Those particles are considered as a human for the estimation of social forces. A normal behaviour of social force model is learned during the training phase using bag-of-words. A test sample is declared anomalous or normal based on the number of matches found from the bag-of-words database.

Like Mehran, Wu *et al.* [61] also used particle advection to model motion behaviors in order to detect anomalies and localize them in a video frame. However, unlike Mehran's approach, the number of grid points are kept the same as the number of pixels. Trajectories of advected particles are clustered using iterative K-Means and for each cluster a feature set is computed using two invariants: the Lyapunov exponents [60] and the correlation

dimensions [24]. The mean of representative trajectory locations is also included into the feature vector to keep spatial information. A model of normal motion behaviour is learned using a mixture of Gaussians model [52] and then maximum likelihood is used to classify a region as anomalous or normal. A panic is localized by analyzing the distribution of motion trajectories with respect to the learned trajectories.

Many of above methods [32, 37, 61] attempt to model crowd behaviors using statistical methods such as Gaussian distribution, correlations, etc. The main limitation of such methods are that finding a good parametric approximation for a training sample is difficult for real situations, and the difficulty worsens with the increase in the dimension of feature vectors. The other type of method is to fit training data using a non-parametric model [51] (Sec. 2.3.2) such as kernel density functions [44], K-mean [61]. A non-parametric model usually does not make any assumption about training samples, hence they appear more practical; however, they need a large number of training samples for high dimensional sample vectors [30], which makes them impractical for many real purposes.

Recently proposed dictionary learning [29] (Sec. 2.3.3) determines a set of dictionary elements $\hat{\mathbf{d}}$ which can reconstruct a training sample $\mathbf{f}$ by a linear combination of DEs (Eq. 2.11). Dictionary learning does not make any assumption about the training data and unlike the non-parametric model, it does not require a large number of samples for high dimensional data. Due to all of these advantages, dictionary learning appears to be a good choice to model normal behaviors in a panic detection algorithm. For a given test sample $\hat{\mathbf{f}}$, the reconstruction error ρ is a scalar, defined as

$$\rho = |\mathbf{f} - \hat{D}\boldsymbol{\eta}| \tag{2.19}$$

where the coefficient vector $\boldsymbol{\eta}$ is computed for the test sample $\mathbf{f}$ using $\hat{D}$. The characteristics of reconstruction error and the coefficients with respect to the learnt values represent a normal behaviour [16].

Cong *et al.* [16] use multi-scale histogram optical flow (MHOF) in a panic detection system, where MHOF is created by concatenating directional and magnitude information of motion. The whole frame is grided into small fixed cells, and the MHOF of each cell is concatenated to form a full feature vector for the video frame. A set of normal frames are chosen for training purposes and dictionary learning is used to select dictionary elements from feature vectors of the training set. Reconstruction cost is computed for a test case and a high reconstruction cost implies a higher possibility of anomalies. The paper has also proposed a technique to update dictionary elements in real-time.

2.5 Accuracy measurement

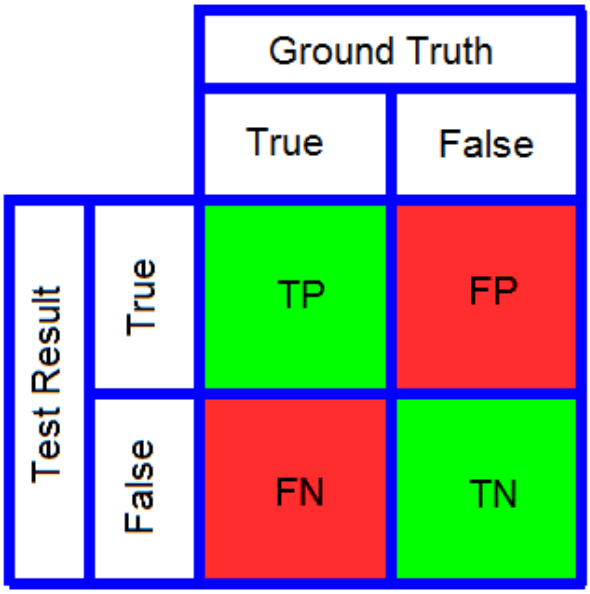


Figure 2.9: There can be two types of outcomes of a classification, namely correct classifications and incorrect classifications. True positive (TP) and true negative (TN) are correct classifications. False negative (FN) and false positive (FP) are wrong classifications. In the case of a panic detection algorithm, panic is considered as positive.

The accuracy of a classifier is usually measured in terms of the F1-measure [35] or the receiver operating characteristic (ROC) [57]. The F1-measure tests the accuracy of a classifier and the ROC (Fig. 2.10) measures the performance of a classifier in terms of a discriminative threshold Γ. The distribution of the F1-measure with respect to Γ provides an optimal value of Γ and the ROC allows us to quantitatively compare the stability of different approaches with respect to Γ.

Figure 2.9 demonstrates a graphical explanation of true positive (TP), false positive (FP), true negative (TN) and false negative (FN) cases in a classifier. In a panic detection method, panic is considered as positive and normal is considered as negative. The F1-measure is defined as

$$\text{F1-measure} = 2\frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \tag{2.20}$$

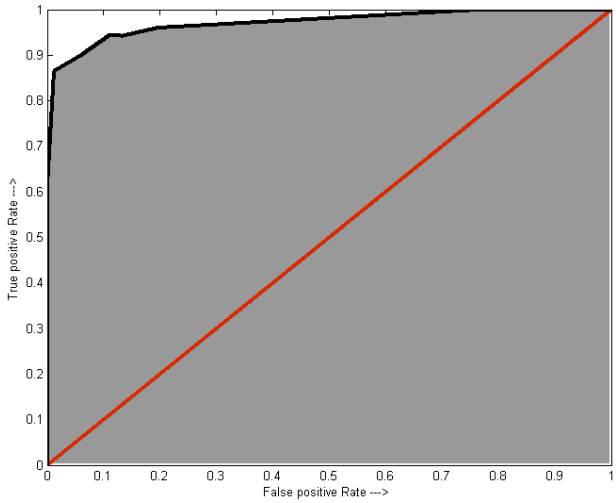


Figure 2.10: A demonstration of an ROC curve. The black curve represents the ROC curve and the red diagonal line represents the accuracy of a perfectly random system. For any binary classifier, the ROC curve must be above red line. The area under ROC (the gray region) is used to compare the accuracy of a classifier.

where, precision $= \frac{TP}{TP+FP}$ and recall $= \frac{TP}{TP+FN}$. An ROC curve is a plot with true positive rate (TPR) on the y-axis and false positive rate (FPR) on the x-axis. Where TPR and FPR are defined as

$$\text{FPR} = \frac{FP}{TN+FP}$$
$$\text{TPR} = \frac{TP}{TP+FN}$$

The area under an ROC curve (AROC) (fig. 2.10) is a measure of the stability for a binary classification system and it is used to compare the accuracy of different classifiers.

2.6 Conclusion

Existing algorithms on panic detection show that for a dense crowd, panic detection becomes challenging on using an object model (Sec. 2.4.1) and a particle model (Sec. 2.4.2) may give promising results. The motion estimate is an important parameter for a good panic detection system. Usually normal behaviour models require high dimensional data,

and dictionary learning (Sec. 2.3.3) appears to be a versatile solution for modeling of such a high dimensional data. Dictionary learning iteratively finds optimal dictionary elements using a set of training samples and in each iteration it uses an operator such as OMP (Sec. 2.3.3) to obtain an optimal coefficient vector $\boldsymbol{\eta}$. OMP determines an optimal set of DEs out of a fixed set of DE. This makes OMP computationally less expensive in comparison with dictionary learning. High accuracy has been obtained with dictionary learning based panic detection systems [16, 63] and it will be useful to know if a relatively inexpensive, OMP based panic detection can also give a competitive accuracy. Chapter 3 develops an algorithm to use OMP for panic detection and Chapter 4 tries to quantitatively answer a few research questions raised at page 46.

Chapter 3

Methodology

As introduced in Chapter 2, this thesis is proposing to develop a simplified model for panic detection in a human crowd using orthogonal matching pursuit (OMP) [47]. An anomaly in a human crowd can encompass many types of abnormalities (discussed at page 1) and a panic includes a subset of an anomaly in the crowd defined in terms of motion irregularities (explained at page 3). A normal behaviour in a crowd corresponds to a case when people walk with their typical speed; people can move coherently such as in an escalator or randomly such as in a park. A panic situation appears when the motion characteristic of a group of people suddenly changes to an previously unobserved behaviour. Our panic detection algorithm is composed of four main segments:

- Preprocessing (Sec. 3.1)
- Training (Sec. 3.2)
- Temporal localization (Sec. 3.3)
- Spatial localization (Sec. 3.4)

3.1 Preprocessing

A panic involves a significant change in the motion behaviour, but the change does not appear to be in a coherent fashion, hence motion direction may not be an important feature for a panic detection method. Moreover, a typical motion estimation method provides both

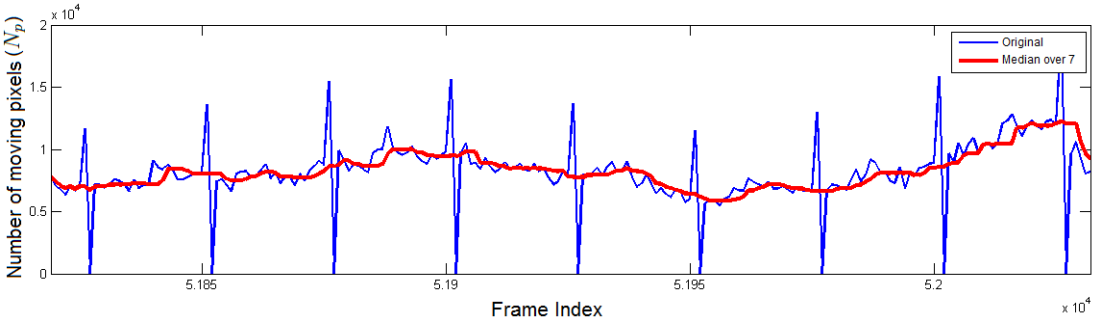


Figure 3.1: The blue curve shows temporal variations of the total number of moving pixels N_p (Eq. 3.2, $\nu = 0.2$) in a sample video. Many small variations are seen due to the presence of erroneous changes in the motion pattern. Large spikes in equal intervals appear because the video frames are not equally distributed in time[1]. The red curve shows the result of median filter on N_p with a window size seven.

direction and magnitude of motion and they are computationally expensive. We think that if the requirements of an algorithm can be restricted to only motion magnitude, a relatively inexpensive motion estimation method can be developed. Hence, the proposed method will use only motion magnitude to check if a panic detection can give a competitive accuracy without motion direction. The motion magnitude $\mathbb{m}(\mathbb{x})$ at position $\mathbb{x}$ is defined as

$$\mathbb{m}(\mathbb{x}) = \sqrt{u(\mathbb{x})^2 + v(\mathbb{x})^2} \tag{3.1}$$

where $u(\mathbb{x})$ and $v(\mathbb{x})$ are x and y components of motion map at position $\mathbb{x}$.

To do a primitive analysis of the motion estimation, the total number of moving pixels N_p is calculated based on the motion estimate [11] as

$$N_p = \sum_{\forall \mathbb{x} \in \mathbb{C}} |\mathbb{m}(\mathbb{x}) > \nu|_0 \tag{3.2}$$

where ν is a threshold to filter out small motion values. The variable N_p is plotted for a video sample through time. The plot (Fig. 3.1) shows a significant amount of disturbance in N_p, these disturbances are found primarily due to unequal temporal differences between consecutive frames in a few video samples[1]. To suppress these unwanted disturbances a median filter of seven elements is used, the red curve in Fig. 3.1 shows the output of the median filter on the erroneous actual data.

[1]It seems that for a few sample videos, video frames are not recorded at equal time intervals. The time gap between two consecutive frames appears to be abnormally high at equal intervals, and then to compensate the high time difference next frame is same as the previous frame.

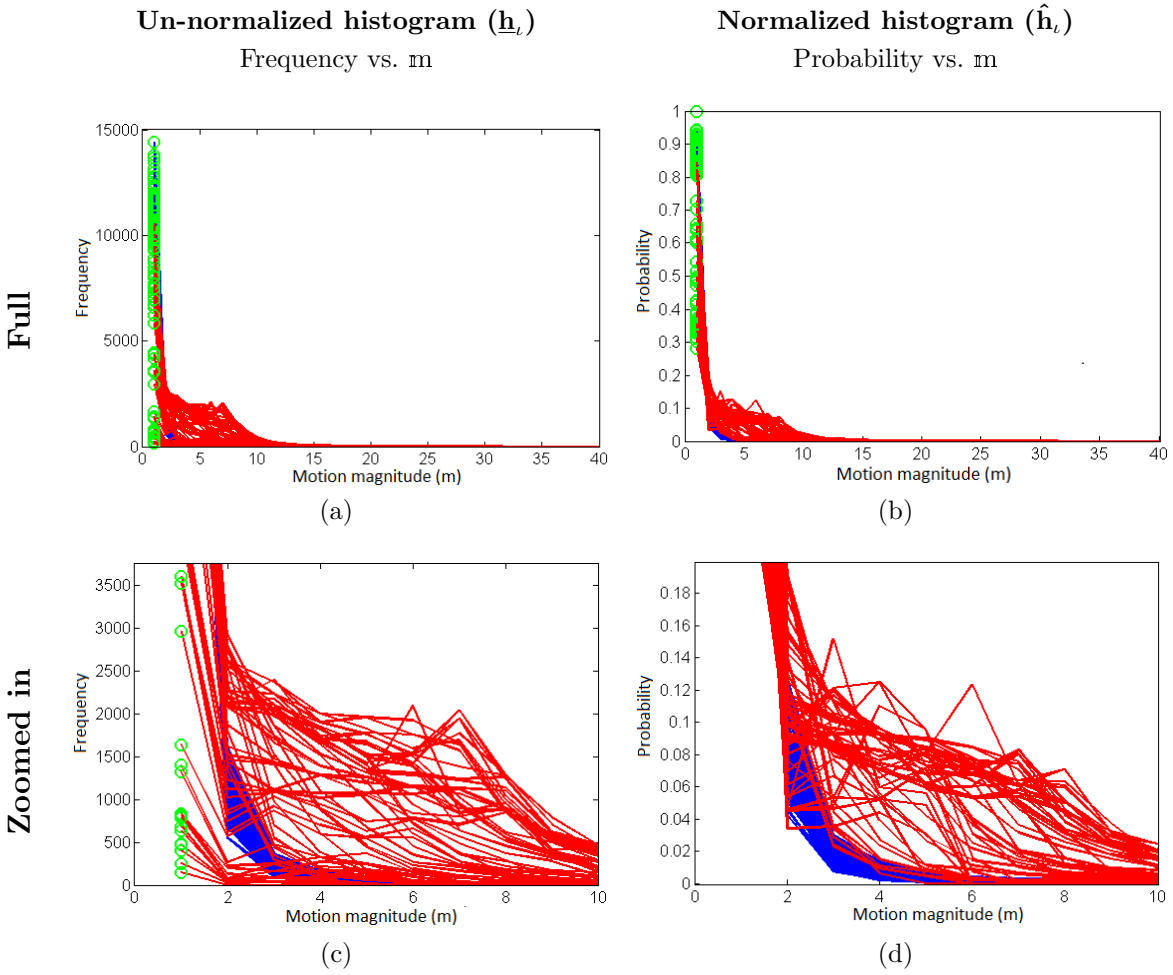


Figure 3.2: (a) and (b) plot all the un-normalized HOFs $\underline{\mathbf{h}}_\iota$ and normalized HOFs $\underline{\mathbf{h}}_\iota$ respectively for a given sample using constants $\nu = 0.02$ and $\zeta = 40$. Green circles help in visualizing histograms behaviors by highlighting the beginning of the each histogram. It shows the variation of histogram for normal and abnormal behaviors. Based on the ground truth, blue curves correspond to normal behaviors and red curves correspond to panics. (c) and (d) zooms in the bottom left corner of (a) and (b) respectively. For normalized histogram, the blue curves stay distinctly apart from red curves near the bent, so HOFs in (d) is more consistent for the normal behaviors as compared to HOFs during a panic.

For an accurate detection of panic, the motion characteristics of all the normal frames should look similar to each other and it should change in the presence of a panic. For a

given frame ι, the histogram of optical flow (HOF) is represented as $\underline{\mathbf{h}}_\iota \in \mathbb{R}^\zeta$, where ζ is the number of bins in a HOF. To avoid saturation due to a large number of stationary pixels, the HOF is defined as

$$\underline{\mathbf{h}}_\iota = \text{histogram}\left(\left\{\mathrm{m}_\iota(\mathbb{x}) \quad \forall \mathbb{x} \in \mathbb{C} | \mathrm{m}_\iota(\mathbb{x}) \geq \nu\right\}\right) \tag{3.3}$$

Each bin of $\underline{\mathbf{h}}_\iota$ contains the total number of pixels corresponding to that bin value. Depending on the total number of moving objects, $\underline{\mathbf{h}}_\iota$ may change significantly. To avoid inconsistencies due to the number of moving objects, the histogram is normalized as

$$\hat{\mathbf{h}}_\iota = \frac{\underline{\mathbf{h}}_\iota}{\sum_{i=1}^{\zeta} \underline{h}_{\iota,i}} \tag{3.4}$$

where $\hat{\mathbf{h}}_\iota$ is a normalized histogram of optical flow. Fig. 3.2 shows that for normal behaviors, the normalized histograms (Fig. 3.2b) stay confined within a small region and in the case of panic they shows high variations. Due to the varying number of moving pixels, the non-normalized HOFs (Fig. 3.2a) do not show clear difference between normal behaviors and panic. Hence, $\hat{\mathbf{h}}_\iota$ appears to be a better choice to model normal behaviour. A distinct pattern during a panic is expected in the normalized histogram $\hat{\mathbf{h}}_\iota$ because in the case of panic, a high proportion of pixels move with abnormally higher speed and hence a significant fraction of the area under a histogram moves toward a higher speed region.

To minimize the effect of noise as observed in Fig. 3.1, the point-wise median of histograms is used as a feature vector. At any frame ι, the corrected histogram $\mathbf{h}_\iota$ is estimated by taking the point-wise median of last w (Set as 7) histograms as

$$h_{\iota,i} = \text{median}\left(\left\{\hat{h}_{j,i} | \forall j \in [(\iota - w), \ \ldots, \iota]\right\}\right) \quad \forall i \in [1, \ \ldots, \zeta] \tag{3.5}$$

Fig. 3.4a shows coefficients corresponding to a learnt dictionary for a given sample video, and it shows many sudden variations due to noise in the sample. On applying median filter corrected histogram ($\mathbf{h}$), the coefficients become smoother (Fig. 3.4b); hence, $\mathbf{h}$ is a better representative of sample behaviors.

3.2 Training

The feature vector h appears consistent during normal behaviors (Fig. 3.2b), hence it is used to model normal behaviors. A histogram is a ζ-dimensional vector and dictionary

learning [29] (Sec. 2.3.3) has been successfully used [16] to model high dimensional feature vectors. Like dictionary learning, wavelet based OMP [47] also gives a set of DEs for a feature vector (Sec. 2.3.3) but the approach is relatively less expensive; therefore, wavelet based OMP is used on N_s training histograms h to extract dictionary elements. Almost all training histograms are found to give the same set of dictionary elements, which shows that these dictionary elements can potentially represent a normal behaviour in a human crowd. A dictionary $\hat{D}$ is formed using N_M dictionary elements found based on training.

The dictionary $\hat{D}$ is used to estimate coefficients (Eq. 2.13) and reconstruction errors (Eq. 2.19) for a training histograms h. For most of the training samples, the reconstruction error ρ (Eq. 2.19) is found to be less than 1% (Fig. 3.3b) compared to $||h||_2$, which shows the selected dictionary $\hat{D}$ is sufficient to model a normal behaviour. The distribution of coefficients η and reconstruction error ρ is not gaussian so median and median absolute deviation (MAD) [50] are used to represent normal behaviors in terms of $\boldsymbol{\eta}$ and ρ. The median of reconstruction errors ($\bar{\rho}$) and the MAD of reconstruction errors (σ_ρ) are defined as

$$\bar{\rho} = \text{median}(\{\rho_i \quad \forall i \in [1, \ \dots, N_s]\}) \tag{3.6}$$

$$\sigma_\rho = \text{median}(\{\rho_i - \bar{\rho} \quad \forall i \in [1, \ \dots, N_s]\}) \tag{3.7}$$

The median of coefficient ($\bar{\boldsymbol{\eta}}$) and the MAD of coefficient ($\boldsymbol{\sigma_\eta}$) are defined as

$$\bar{\boldsymbol{\eta}} = \text{median}(\{\boldsymbol{\eta_i} \quad \forall i \in [1, \ \dots, N_s]\}) \tag{3.8}$$

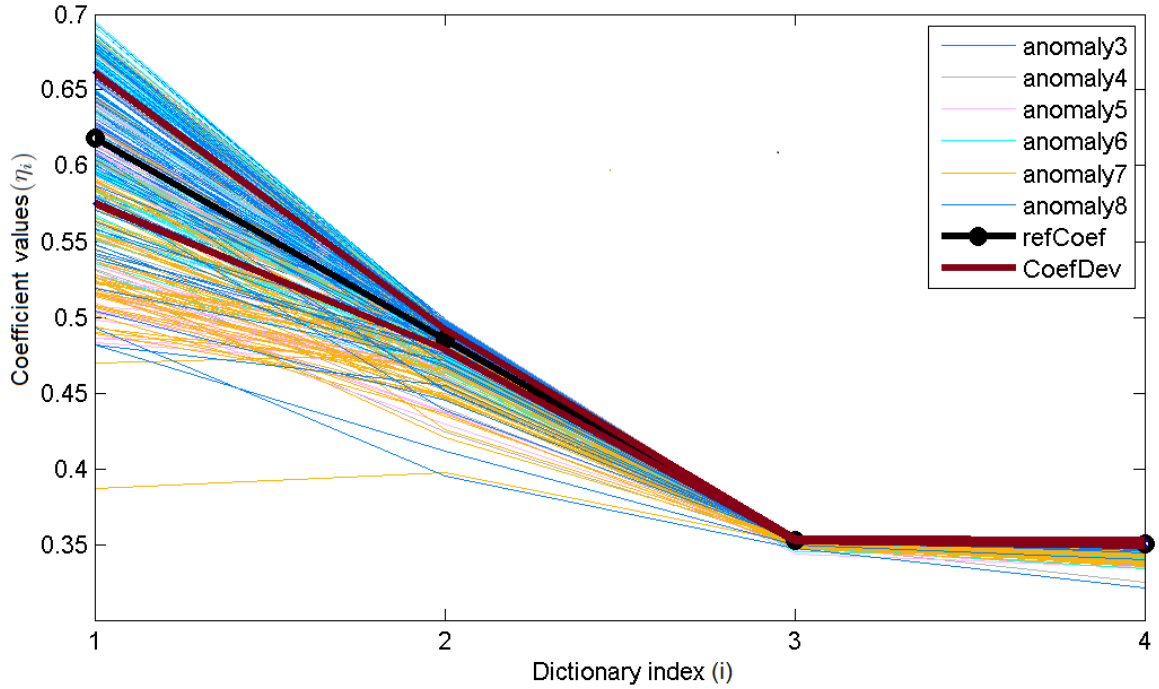
$$\boldsymbol{\sigma_\eta} = \text{median}(\{\boldsymbol{\eta_i} - \bar{\boldsymbol{\eta}} \quad \forall i \in [1, \ \dots, N_s]\}) \tag{3.9}$$

Thus, training provides a dictionary $\hat{D}$, a median of coefficients $\bar{\boldsymbol{\eta}}$, a MAD of coefficients $\boldsymbol{\sigma_\eta}$, a median of reconstruction errors $\bar{\rho}$, and a MAD of reconstruction errors σ_ρ. These features are used in the temporal localization of panic events.

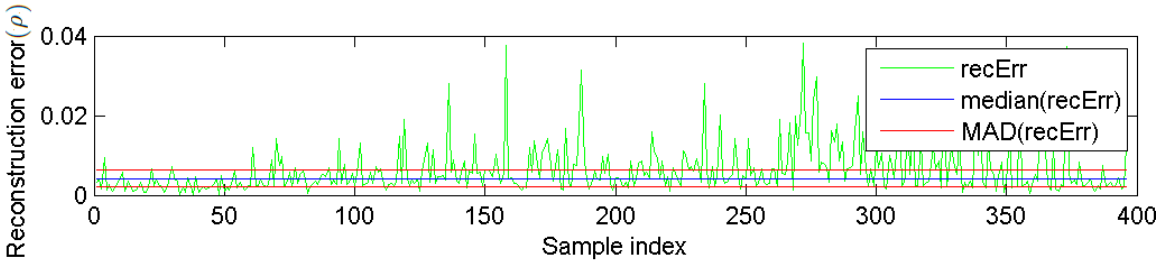
3.3 Temporal localization

In Fig. 3.5, where red regions show panic frames based on ground truth, coefficients corresponding to different dictionary elements change significantly during a panic. This happens because during the panic histogram concentration moves from one place to another, which leads to a significant reduction of the coefficients which were representing high concentration of motion during normal behaviors.

Based on the learnt dictionary $\hat{D}$ and other variables ($\bar{\boldsymbol{\eta}}$, $\boldsymbol{\sigma_\eta}$, $\bar{\rho}$, σ_ρ), a panic can be detected in the following two situations:

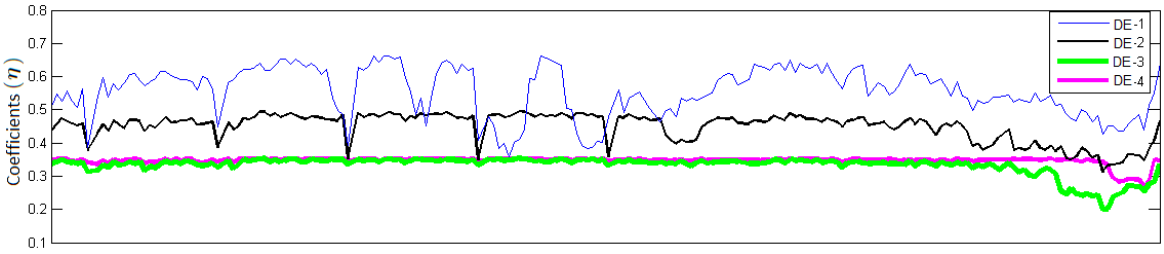


(a) Coefficient values for each dictionary element

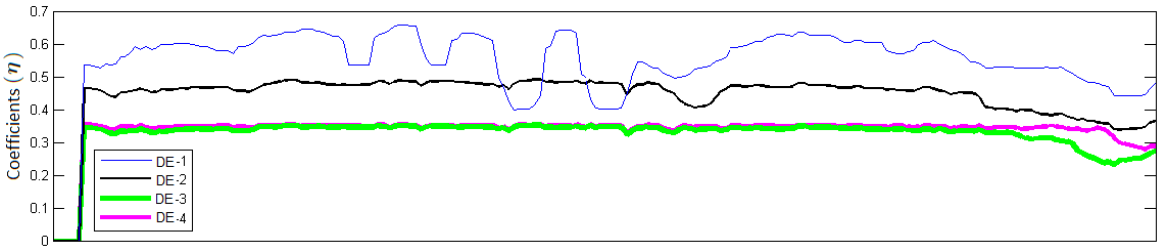


(b) Reconstruction error

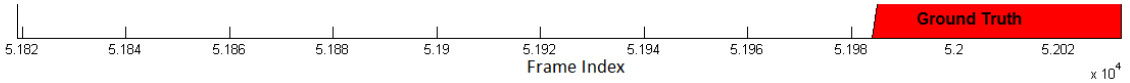
Figure 3.3: Displaying behaviors of wavelet OMP-based dictionary $\hat{D}$ for a training set. A dictionary $\hat{D}$ of four dictionary elements is formed after training with $N_S = 400$ training samples. (a) Coefficients of $\hat{D}$ when used on same training samples. Coefficients corresponding to different dictionary elements show significantly varying degree of variations, for example DE-3 has considerably small variation than DE-1. This observation leads to an improved detection system (Eq. 3.17). (b) Reconstruction errors (Eq. 3.10) estimated using $\hat{D}$ on the training set.



(a) Coefficients without preprocessing filter



(b) Coefficients with filter (Eq. 3.5)



(c) Truthed panic frames (red)

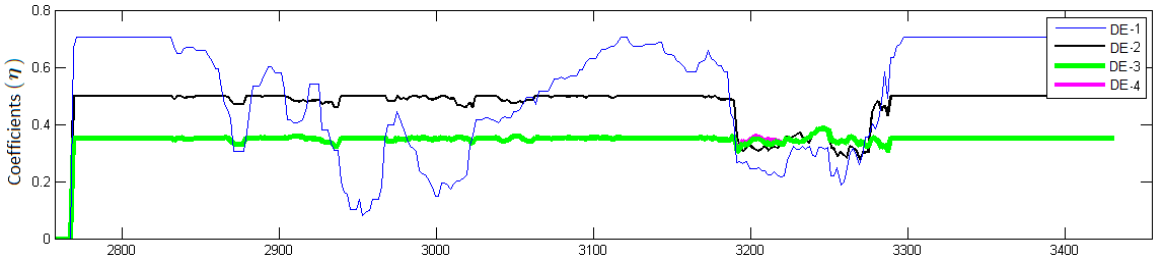
Figure 3.4: (a) and (b) show coefficient values corresponding to each DE for a video sample, plotted in the temporal order; each curve corresponds to one DE, and the thickest curve implies the lowest variance ($\boldsymbol{\sigma_\eta}$) and the highest stability. Coefficient values $\boldsymbol{\eta}$ corresponding to uncorrected histograms $\hat{\mathbf{h}}$ (a) show many erroneous fluctuations. (b) shows reduced variations on using filtered histogram $\mathbf{h}$. The red region shows the panic frames based on the ground truth.

1. The dictionary $\hat{D}$ fails to effectively represent a sample HOF $\mathbf{h}$, and gives an abnormally high reconstruction error ρ. The difference of reconstruction error $\delta\rho$ is used to check a panic and it is defined as

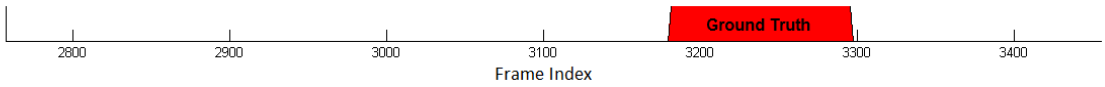
$$\delta\rho = |\rho - \bar{\rho}| \tag{3.10}$$

2. The coefficient error $\delta\eta$ is significantly high, where $\delta\eta$ is expressed as

$$\delta\eta = ||\boldsymbol{\eta} - \bar{\boldsymbol{\eta}}||_2 \tag{3.11}$$



(a) Coefficients



(b) Truthed frame index

Figure 3.5: Coefficient values η_i corresponding to i^{th} DE, computed using the wavelet OMP based dictionary $\hat{D}$, are plotted through time in (a). The red region in (b) shows panic frames based on the ground truth. In (a) the thinnest curve (blue) represents a DE with the highest variance and minimum stability, computed during the training (Fig. 3.3a). The DE (DE-1) which shows high variations on the training set also shows high variations on the test frames with normal behaviors. The most stable DE (DE-3) shows almost no variation during normal behaviors.

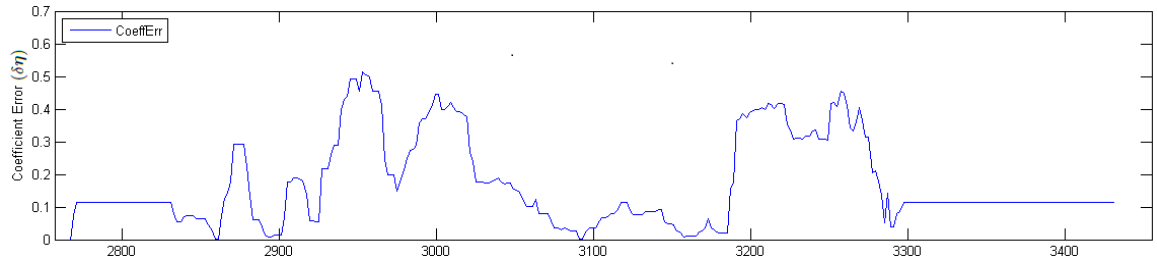
Fig. 3.6 shows that during a panic $\delta\rho$ and $\delta\eta$ increase significantly; however, there are a few other frames also where these parameters show high values in an uncorrelated manner, these unwanted high values occur because of variations in human motion. Errors are also introduced because a video frame does not contain depth information. In the absence of depth information, motion magnitudes are heavily dependent on the motion direction. A person walking along the camera axis appears to walk at a slower pace than a person walking (with the same speed as previous one) perpendicular to the camera axis. For accurate detection, it is important to have a parameter which reliably shows a high difference during a panic. As $\delta\rho$ and $\delta\eta$ both rise concurrently during a panic, they are combined to obtain the joint error $\mathbb{E}$ defined as

$$\mathbb{E} = \delta\eta + \beta\delta\rho \tag{3.12}$$

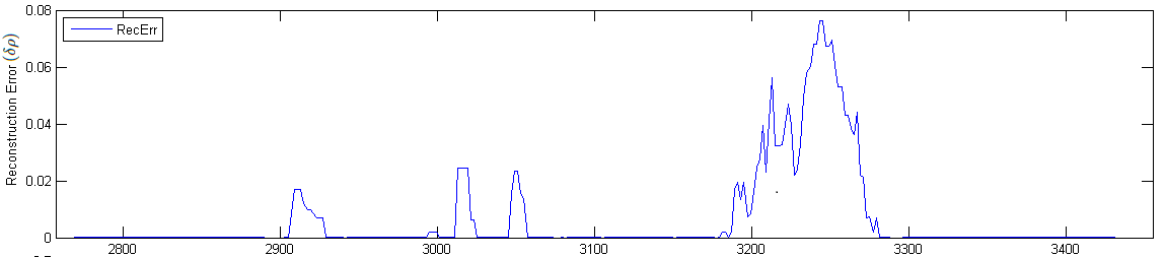
where β is a constant to set the relative weight of $\delta\rho$ and $\delta\eta$. The joint error $\mathbb{E}$ is compared with a threshold Γ to detect a panic as

$$\text{Panic} = \begin{cases} \text{true} & \text{if } \mathbb{E} \geq \Gamma \\ \text{false} & \text{otherwise} \end{cases} \tag{3.13}$$

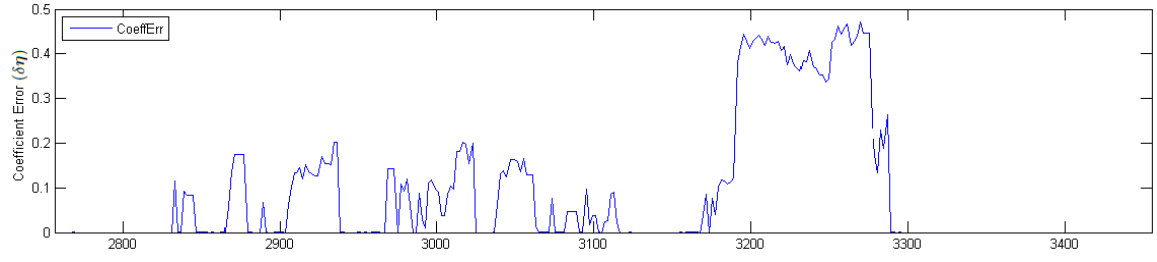
Fig. 3.7a shows detected panic frames for a video sample. In the plot, $\mathbb{E}$ shows two high



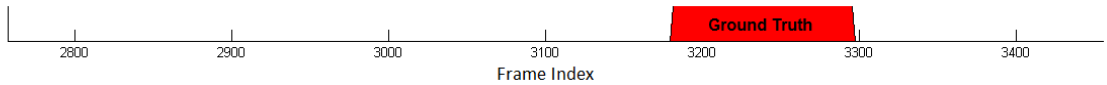
(a) Coefficient error ($\delta\boldsymbol{\eta}$) (Eq. 3.11)



(b) Difference of reconstruction error ($\delta\rho$)(Eq. 3.10)



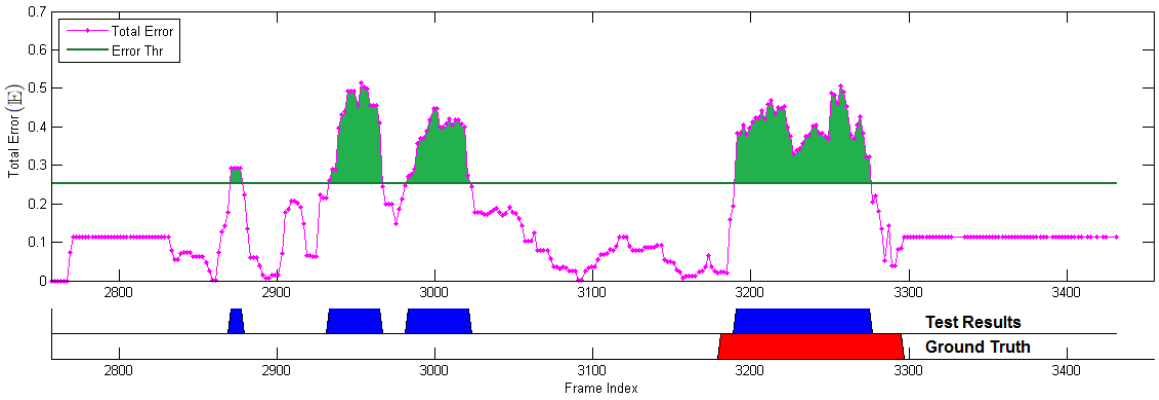
(c) Coefficient error ($\delta\boldsymbol{\eta}$) (Eq. 3.17)



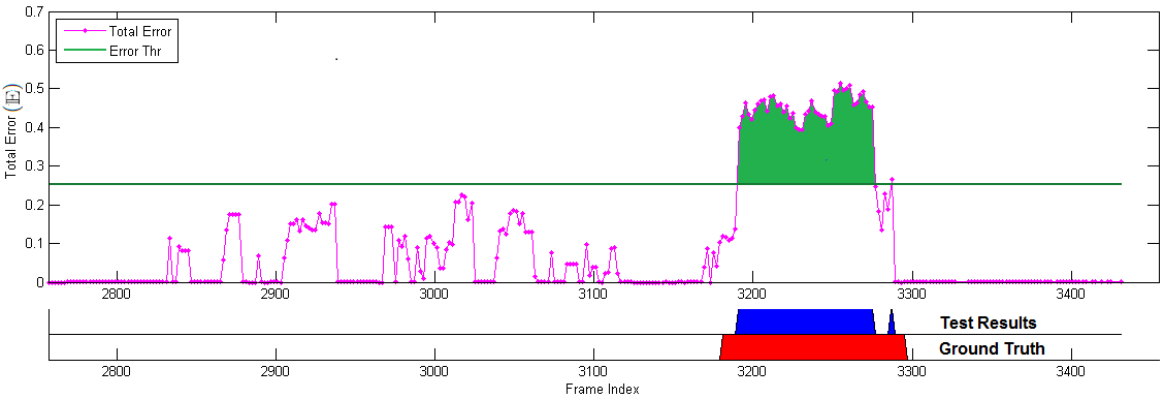
(d) Truthed frame index

Figure 3.6: Coefficient errors ($\delta\eta$) are shown in (a) and (c)corresponding to the coefficients (Fig. 3.5) for a test sample. Red regions in (d) show panic frames based on ground truth. Coefficient errors (a) using Eq. 3.11 show high disturbances even in normal situations but coefficient errors (c) estimated using Eq. 3.17 give distinctly high peaks during a panic. Reconstruction errors ($\delta\rho$), plotted in (b), also show large changes during a panic.

peaks during normal behaviors, these peaks are almost as high as the peak for the panic situation. Due to these unwanted peaks many false positive panic detections are observed.



(a) Total error (𝔼) using coefficient error (Eq. 3.11)



(b) Total error (𝔼) using coefficient error (Eq. 3.17)

Figure 3.7: Test results using total error $\mathbb{E}$ (Eq. 3.12), the coefficient error has been computed using coefficients shown in Fig. 3.5. (a) Coefficient errors using Eq. 3.11 (Approach I) and (b) Coefficient errors using Eq. 3.17 (Approach II). Green-filled regions show frames where $\mathbb{E}$ is found higher than the discriminative constant Γ (shown by the green line). In both (a) and (b), blue regions show detected panic frames based on Eq. 3.13 and red regions show ground truth. Approach II gives a more accurate estimation because it gives low weight to unstable DEs, thus not allowing unstable DEs to significantly affect $\mathbb{E}$.

The following paragraphs explain a weighted method of coefficient error computation to improve the accuracy of the system.

Coefficients are computed for all N_M training samples using the learnt dictionary $\hat{D}$. Fig. 3.3a shows that the coefficient for dictionary element 1 shows higher variations than

the coefficient for dictionary element 3. A low variation in a dictionary element implies that it is a stable representative of the training sample and vice-versa. A large variation in the stability of DEs demands that coefficient errors corresponding to different DEs should be treated differently. Coefficient errors corresponding to a stable DE should be penalized more than those for a less stable DE. If $\sigma_{\eta,i}$ represents the i^{th} element in $\boldsymbol{\sigma_\eta}$, then the highest value of $\sigma_{\eta,i}$ implies the lowest penalty. One way of achieving this is by modifying Eq. 3.11 as

$$\delta\eta_i = \frac{\eta_i - \bar{\eta}_i}{\sigma_{\eta,i}} \quad \forall i \in [1,\ \dots, N_M] \tag{3.14}$$

where $\delta\eta_i$ is the i^{th} element of the coefficient vector error. It is possible that $\sigma_{\eta,i}$ will be close to zero, which will lead to a very high value in $\delta\eta_i$, hence Eq. 3.14 is not a desirable formulation. A multiplying vector $\boldsymbol{\Psi} \in \mathbb{R}^{N_M}$ appears to be a better option. The i^{th} element of the vector $\hat{\boldsymbol{\Psi}}$ is multiplied with $(\eta_i - \bar{\eta}_i)$ such that the DE with lowest $\sigma_{\eta,i}$ can contribute most and vice-versa.

A vector $\boldsymbol{\phi} = [\phi_1,\ \dots, \phi_{N_M}]$ is defined such that it contains indices of $\boldsymbol{\sigma_\eta}$ in a sorted order $\boldsymbol{\sigma_\eta}(\phi_1) < \boldsymbol{\sigma_\eta}(\phi_2) < \ \dots\ < \boldsymbol{\sigma_\eta}(\phi N_M)$, where $\boldsymbol{\sigma_\eta}(\phi_j)$ is the ${\phi_j}^{th}$ element of $\boldsymbol{\sigma_\eta}$. The j^{th} element of the weight vector $\boldsymbol{\Psi}$ is defined as

$$\Psi_j = \boldsymbol{\sigma_\eta}(\phi_{N_M - j + 1}) \tag{3.15}$$

The weight vector $\boldsymbol{\Psi}$ is normalized as

$$\hat{\Psi}_j = \frac{\Psi_j}{\sum_{i=1}^{N_M} \Psi_j} \tag{3.16}$$

Coefficient error for j^{th} DE $\delta\eta_j$ is computed using $\hat{\Psi}_j$ as

$$\delta\eta_j = \hat{\Psi}_j(\eta_j - \bar{\eta}_j) \quad \forall j \in [1,\ \dots, N_M] \tag{3.17}$$

After using $\hat{\boldsymbol{\Psi}}$ in coefficient error, a significant improvement is observed in the panic detection accuracy (as will be discussed in Ch. 4). The weight vector $\hat{\boldsymbol{\Psi}}$ helps in the following ways:

1. Less stable DEs typically capture small variations in the training set and a system becomes less invariant to small variations by using smaller weights associated with less stable DEs.

2. A stable DE stays stable (Fig. 3.5a) unless there is a significant change and thus it improves the stability of a system.

3. The number of unstable dictionary elements depends on the value of ξ (Eq. 2.15). With the proposed weighting mechanism, the most stable dictionary element is weighted proportional to the variation of the most unstable dictionary element and vice- versa. This makes the system accuracy less dependent on ξ.

Based on the two variations of Eq. 3.12, panic detection is defined using two approaches:

Approach I: *This method uses coefficient error computation using Eq. 3.11.*

Approach II: *Eq. 3.17 will be used for the estimation of coefficient error, all other settings shall remain the same as Approach I.*

Approach II shows a significant improvement over Approach I (Fig. 3.6c) in the distribution of $\delta\eta$. Unlike $\delta\eta$ with Approach I (Fig. 3.6c), a distinctly high peak is obtained for panic frames with approach II. The coefficient error was computed on coefficients shown in Fig. 3.5, where the least stable DE (shown by thinnest curve) incurs high fluctuations, whereas the most stable DE (the thickest curve) shows high values only during a panic. On giving low weight $\hat{\Psi}_i$ to less stable DE, the unwanted fluctuations of less stable DEs do not affect $\mathbb{E}$ significantly. A threshold Γ is applied on $\mathbb{E}$, computed based on approach I and II. Figures 3.7a and 3.7b show that the Approach II gives higher accuracy, when a threshold Γ is applied to $\mathbb{E}$, computed based on Approach I and II respectively.

3.4 Spatial localization

Once a panic is temporally localized, spatial localization is done in the panic containing video frame. The histogram of flow magnitude $\mathbf{h}$ gives a distribution of motion across a frame and a similar distribution is expected for all normal frames. In the case of panic, a few pixels move faster than normal and this change is also reflected in the histogram, where, the number of pixels associated with a particular motion magnitude increases. The difference of the average of training histograms $\bar{\mathbf{h}}$ and a test histogram $\mathbf{h}_\iota$ tells us for which motion magnitude there has been a significant change in the number of associated pixels. By finding motion magnitudes, which have been affected significantly, corresponding region in the flow map is found and thus panic is spatially localized.

A average histogram $\bar{\mathbf{h}} \in \mathbb{R}^\zeta$ is estimated by taking the element-wise average of all the training histograms as

$$\bar{h}_i = \frac{\sum_{j=1}^{N_s} h_{i,j}}{N_s} \quad \forall i \in [1, \ \dots, \zeta] \tag{3.18}$$

(a) (b)

Figure 3.8: (a) a panic frame of a sample video and (b) the spatial localization of panic for (a) using constant $\upsilon = 2$ and $\varphi = 0.001$. Green regions in (b) show the spatial localization of panic (Sec. 3.4).

A standard deviation of each element of the histogram ($\boldsymbol{\sigma_h} \in \mathbb{R}^{\zeta}$) is estimated as

$$\sigma_{h,i} = \sqrt{\frac{\sum_{j=1}^{N_s}(h_{i,j} - \bar{h}_i)^2}{N_s}} \quad \forall i \in [1, \ \dots, \zeta] \tag{3.19}$$

The standard deviation helps in deciding the range of allowed difference between the test histogram and the average histogram. For a frame, the histogram error δh is estimated as

$$\delta \mathbf{h} = \mathbf{h} - (\bar{\mathbf{h}} + \upsilon * \boldsymbol{\sigma_h}) \tag{3.20}$$

where υ is a constant. A decrease in motion magnitude does not correspond to a panic, so only positive change in histogram is considered for the spatial localization. A set H of affected motion magnitude is formed using $\delta \mathbf{h}$ as

$$H = \{\delta h_i \mid \delta h_i > \varphi \quad \forall i \in [1, \ \dots, \zeta]\} \tag{3.21}$$

where φ is a constant. In a given image, all those areas are highlighted for which motion magnitude lie within the range of any element of H. Fig. 3.8b shows a frame where a panic is spatially localized and highlighted by green color.

Chapter 4

Results

Chapter 3 explains a novel panic detection algorithm based on OMP [47]. The algorithm is developed to computationally simplify existing algorithms for panic detection and test the effect of simplification on the accuracy. This chapter will try to answer the following important questions quantitatively.

Research questions

1. *Is it possible to develop a high accuracy panic detection system using a wavelet OMP (Sec. 2.3.3)?*

   Dictionary learning [29] is a robust way of modeling normal behaviors; however, it is computationally expensive. Wavelet based OMP is relatively inexpensive (Sec. 2.6) and hence an OMP based panic detection algorithm should also be inexpensive compared to a dictionary learning based system. The accuracy of the proposed algorithm (Ch. 3) has been compared with three other methods:

   (a) *Sparse reconstruction cost for abnormal event detection [16].*

   (b) *Abnormal crowd behavior detection using social force model [43].*

   (c) *Chaotic invariants of Lagrangian particle trajectories for anomaly detection in crowded scenes[61].*

2. *Can only the motion magnitude be used for a panic detection algorithm?*

   Motion estimation is the computationally most expensive part of a panic detection algorithm (Sec. 4.3.4) which makes most of the available detection system non-realtime.

(a) Motion along the camera axis. (b) Motion perpendicular to the camera axis. (c) A running person.

Figure 4.1: Three sample frames from Subway 2 dataset (Table 4.1) are shown. People walking along the camera axis (a) appear to move slower than people walking perpendicularly to the camera axis (b). (c) shows a panic situation where a person is running.

Most of the existing panic detection algorithms have used both motion direction and magnitude. If a competitive accuracy can be obtained with only motion magnitude, a computationally inexpensive flow estimation can be designed to estimate only motion magnitude, which will help in a real-time panic detection method.

3. *What is the impact of flow estimation method on the panic detection accuracy?*

   Usually optical flow [11] is used in a panic detection algorithm. The flow estimated based on the optical flow and SIFT flow [38] are significantly different. SIFT flow is less noisy but a little pixelated (Fig. 2.5d). As both flow estimates have pros and cons, an accuracy comparison with respect to these two flow estimates will help in analyzing the robustness of the proposed method.

4.1 Data sets

Two datasets have been used to test the proposed algorithm.

1. **Unusual crowd activity from the University of Minnesota** [1]: In this dataset, (Fig. 1.1) a number of people walk normally in a given area, and after sometime they start to run randomly and eventually leave the area. The dataset contains a video with 11 panic situations, collected from three different places. Table 4.1 contains names of all samples in the dataset categorized by scenes. The video frames are

tagged with *normal* or *anomaly* at the top-left corner but the tagging does not look to be correct, as panic starts much sooner before the tagging changes its status to *anomaly*. So we truthed the video and the ground truth information has been included with Table 4.1.

2. **Subway dataset from the Technion - Israel Institute of Technology [2]:** The sample contains 5 videos, which include test cases for panic and other types of anomalies. The algorithm has been tested on only the second video (Subway 2) which contains panic. The video has been filmed in a mall, where people are walking normally and suddenly a person runs across the mall (Fig. 4.1c). Running represents a type of panic situation which needs to be detected. The video contains nine cases of panics which have been tested by the proposed algorithm. We created the ground truth and the ground truth information of Subway 2 has also been included with Table 4.1.

   Compared to the University of Minnesota sample, this sample is more difficult to detect panic because in the sample most of the people are walking parallel to the camera axis (Fig. 4.1a) and a few walk perpendicularly to the camera axis (Fig. 4.1b). In a video frame, pixels corresponding to a person walking parallel to the camera axis move slower than pixels corresponding to a person walking perpendicularly to the axis, this tends to lower the accuracy.

4.2 Experimental setup

The algorithm has been tested with two types of flow, optical flow [11] and SIFT flow [38]. Optical flow has been computed using Brox *et al.*'s mex implementation and SIFT flow has been computed with Liu *et al.*'s Matlab implementation[1].

The following parameters have been used in the flow computation, these parameters are set as suggested by the developers of flow estimation methods [11, 38].

- optical flow (Sec. 2.2.1): $\alpha = 60$, $\gamma = 10$.
- SIFT flow (Sec. 2.2.2): p = 8.

[1]Brox *et al.*'s implementation is available at http://www.cs.brown.edu/people/dqsun/, and Liu *et al.*'s code is available at http://people.csail.mit.edu/celiu/ECCV2008/release.zip

A histogram of optical flow $\mathbf{h}$ with dimensions $\zeta = 40$ is used and bins are kept at values $\{1, \ ... \ , \zeta\}$. The magnitude of optical flow represents the motion of a pixel in an image in terms of the number of pixels and experiments shows that pixels do not move by more than 20 pixels for any sample. So, $\zeta = 40$ is used to allow the worst case. The constant ν (Eq. 3.3) is set to 0.2 to filter out stationary pixels and β is set to 1.

Total $N_S = 400$ samples are used during training with OMP. For a given set (Sec. 4.1, Table 4.1) 400 random HOFs are selected corresponding to normal behaviors. Wavelet based OMP is run on them to select representative dictionary elements. The initial dictionary D is formed using reverse bi-orthogonal-1 wavelet [2] [55]. Reverse-bi-orthogonal-1 wavelet is chosen because it fits a HOF properly and it is an orthogonal wavelet. For the spatial localization, the reference histogram (Eq. 3.21) is computed with $\upsilon = 2$ and the threshold ϕ is set to 0.001, these constants are experimentally selected. All of these parameters are kept same while testing with SIFT flow.

4.3 Results analysis

The system is trained for each set and the learned dictionary $\hat{D}$ is used to test samples of that set. Two accuracy measurement schemes, namely AROC [57] and F1-measure [35], are used to compare the accuracy. Most of the existing algorithms on the panic detection have reported and compared results in terms of AROC. So, the accuracy of the proposed algorithm will be compared with others in terms of AROC. AROC gives a good idea about the stability of a system; however, it does not give much information about the accuracy of a system. The F1-measure measures the accuracy of a classifier and hence it has also been reported for completeness of our analysis.

A panic is considered as a positive case and a normal behaviour is considered as a negative case. Both the accuracy measurement schemes need to count the total number of positive and negative cases to compute FP, TP, FN and TN (Sec. 2.5). For the accuracy of the proposed algorithm, the count of positive and negative cases will be reported based on the number of frames. For an example, if 50 frames are truthed positive and 30 overlapping frames are tested positive then $TP = 30$.

The accuracy is computed for all the samples based on dictionary $\hat{D}$ of their corresponding sets. Five different analyses have been presented here:

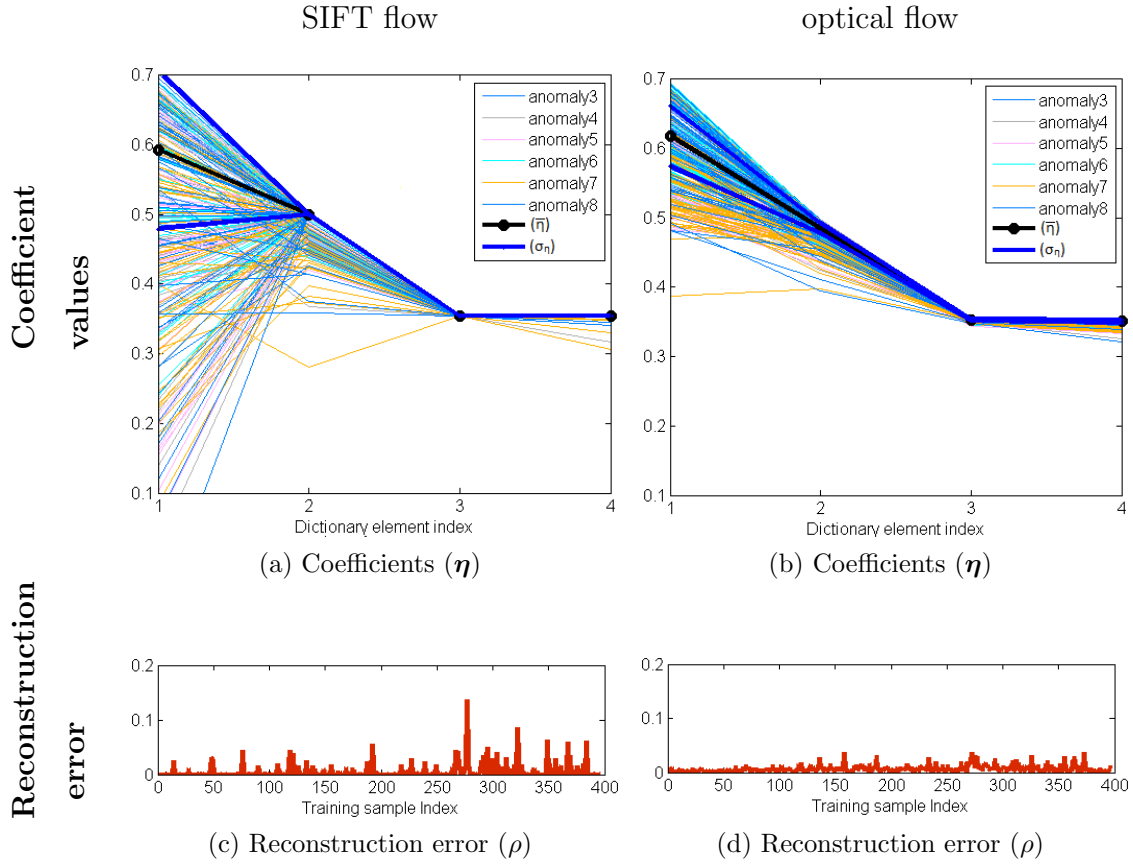
1. SIFT flow vs optical flow (Sec. 4.3.1)

[2] In Matlab this wavelet is defined as rbio1.1

Table 4.1: Details of datasets used are listed here and explained in Sec. 4.1. The last two columns list the ground truth frame indices of each sample video. In subway2-10, there is no panic, it is used only for the training purpose.

Source	Set Name	Sample Name	Start Frame	End Frame	Panic Start	Panic End
Univ. of Minnesota [1]	Scene 1	panic1	1	625	475	617
		panic1	627	1453	1294	1331
	Scene 2	panic3	1455	2000	1773	1961
		panic4	2003	2686	2589	2686
		panic5	2687	3455	3180	3295
		panic6	3456	4019	3920	4019
		panic7	4035	4929	4783	4895
		panic8	4930	5595	5402	5504
	Scene 3	panic9	5597	6253	6131	6253
		panic10	6255	6931	6828	6931
		panic11	6933	7739	7651	7739
Subway dataset [2]	Subway 2	subway2-1	49617	50202	50160	50202
		subway2-2	50204	50966	50834	50875
		subway2-3	50968	51430	51216	51265
		subway2-4	51432	51816	51600	51640
		subway2-5	51818	52032	51985	52032
		subway2-6	52034	52428	52365	52428
		subway2-7	52430	52826	52785	52826
		subway2-8	52828	53346	53143	53240
		subway2-9	53348	53858	53805	53858
		subway2-10	53860	55000		

2. Approach I vs Approach II (Sec. 4.3.2)
3. Accuracy comparison with other approaches (Sec. 4.3.3)
4. Time complexity (Sec. 4.3.4)
5. Spatial localization (Sec. 4.3.5)



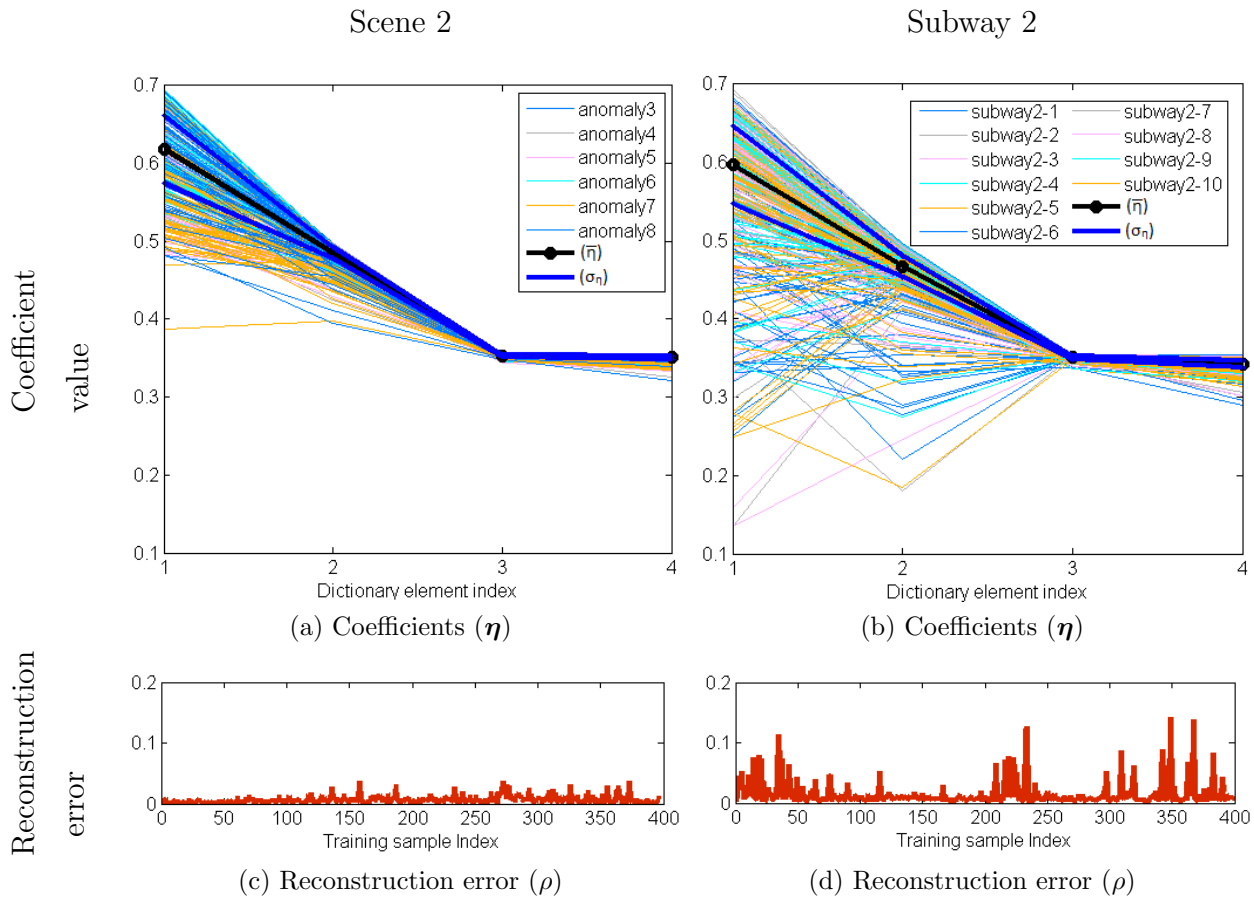
(a) Coefficients ($\boldsymbol{\eta}$) (b) Coefficients ($\boldsymbol{\eta}$)

(c) Reconstruction error (ρ) (d) Reconstruction error (ρ)

Figure 4.2: Coefficients are computed on the training samples using a dictionary $\hat{D}$, learnt on Scene 2 (Table 4.1) using (a) histogram of SIFT flow and (b) histogram of optical flow. Optical flow based coefficients have less variability than SIFT flow based coefficients. Reconstruction errors (Eq. 2.19) computed on those training samples are smaller for optical flow based histogram (d) supporting the observation that optical flow based histograms are better representatives of normal behaviors. Table 4.2 further validates this observation and shows that the AROC of an optical flow based system is a little higher than the AROC of a SIFT flow based system.

4.3.1 SIFT flow vs optical flow

Optical flow appears smoother than SIFT flow (Fig. 2.5) however, it is more noisy. SIFT flow performs better in the case of large displacements. Tests are run on all samples with SIFT flow and optical flow separately to see how these differences of flow affect the



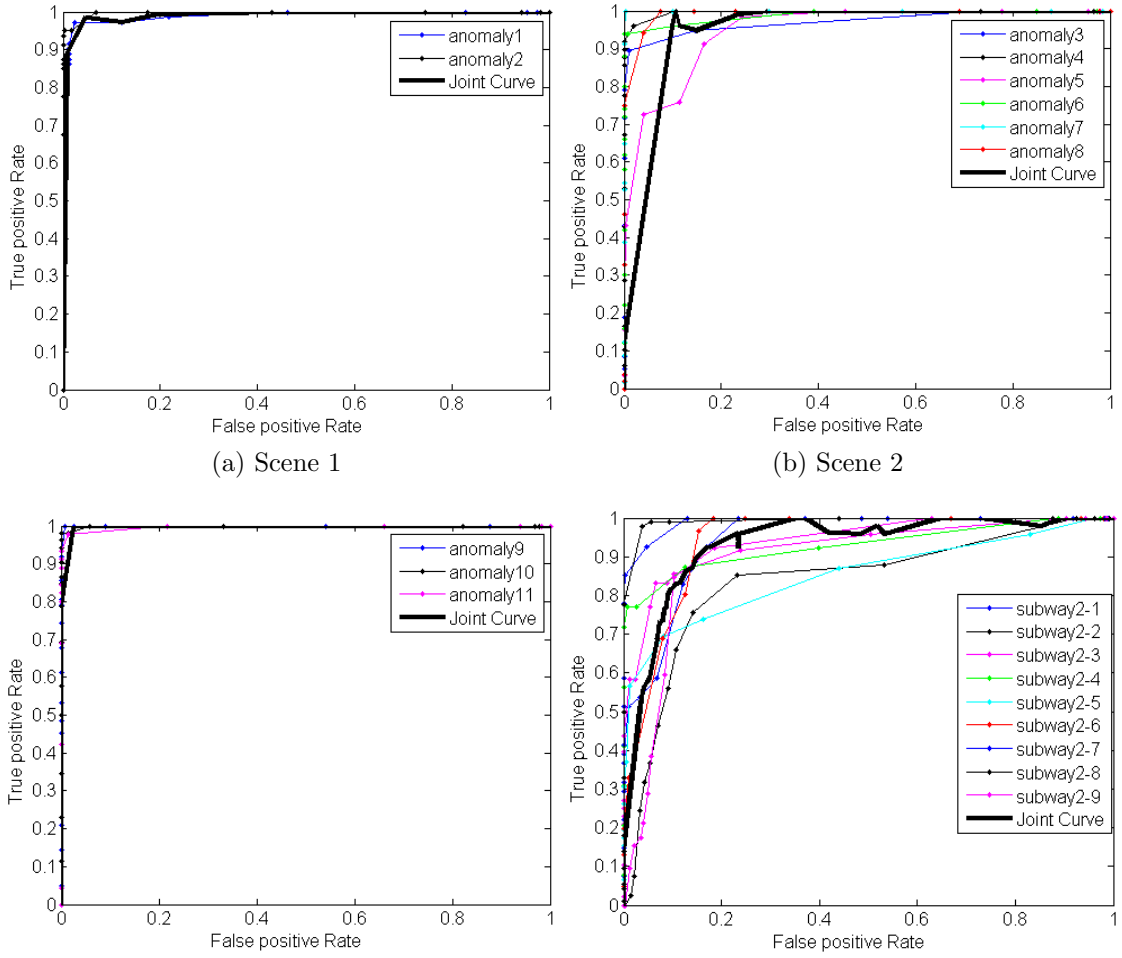
(a) Coefficients ($\boldsymbol{\eta}$) (b) Coefficients ($\boldsymbol{\eta}$)

(c) Reconstruction error (ρ) (d) Reconstruction error (ρ)

Figure 4.3: Coefficients for datasets (a) Scene 2 and (b) Subway 2 (Table 4.1), computed on training samples using a learnt dictionary $\hat{D}$. Coefficients for Subway 2 appear more scattered than coefficients for Scene 2, which implies Subway 2 has higher variation during normal behaviour. This observation is further supported by high reconstruction errors in the case of (d) Subway 2 as compared to (c) Scene 2. The higher variation in normal samples in the Subway 2 dataset causes the lower accuracy in the case of Subway 2 dataset, as included in Tables 4.3 and 4.2.

accuracy. Results are included in Table 4.2.

On comparing AROC, the first two columns or the last two columns, the optical flow based algorithm almost always performs better than the SIFT flow based algorithm. Coefficients $\boldsymbol{\eta}$ are estimated on training sample using $\hat{D}$. In all samples coefficients estimated based on the optical flow show higher compactness then those based on SIFT flow (Fig. 4.2).



(a) Scene 1

(b) Scene 2

(c) Scene 3

(d) Subway 2

Figure 4.4: Each plot shows ROC curve for a sample set. The joint curve (thick black curve) for each sample set (listed in Table 4.1) represents the average ROC curve of all samples in that set using approach I. Other curves represent ROC corresponding to the individual samples.

This shows that optical flow based $\hat{D}$ is a better representative of normal behaviors and hence it can detect panic with higher accuracy. The observation is further verified by the observation that the reconstruction error ρ based on the SIFT flow produces higher value compared to the optical flow. Although optical flow shows higher error in the motion estimation than SIFT flow, erroneous pixels do not seem to affect the overall accuracy. The

F1-measure (Table 4.3) produces a similar comparison pattern as AROC.

Table 4.2: Optical flow gives better accuracy than SIFT flow by a small fraction, this observation is supported by the observed compactness of training coefficients for optical flow as compared to SIFT flow (Fig. 4.2). For the Minnesota dataset, AROC by approach II is lower than approach I; however, approach II improves the accuracy by a significant fraction for Subway 2 dataset. Sec. 4.3.2 explains this observation in detail. Fig. 4.4 shows the ROC for all data sets based on optical flow.

	Approach I		Approach II	
	SIFT flow [38]	optical flow [11]	SIFT flow [38]	optical flow [11]
Scene 1	0.962	0.994	0.959	0.989
Scene 2	0.953	0.974	0.977	0.949
Scene 3	0.996	0.999	0.951	0.997
Subway 2	0.893	0.875	0.926	0.9307

Table 4.3: Due to the higher compactness of the learned coefficients (Fig. 4.2) for optical flow, the F1-measure for the optical flow based system is higher by a small fraction. Compared to approach I, approach II shows improvements for a few sets and regression for a few sets. Approach II gives significant improvements over approach I for Subway 2 because there are a few unstable dictionary elements in the dictionary $\hat{D}$ of Subway 2 and approach II has been developed to handle unstable dictionary elements (Sec. 4.3.2). Fig. 4.5 shows the F1-measure for all the data sets based on optical flow.

	Approach I		Approach II	
	SIFT flow [38]	optical flow [11]	SIFT flow [38]	optical flow [11]
Scene 1	0.968	0.969	0.945	0.952
Scene 2	0.805	0.906	0.911	0.880
Scene 3	0.972	0.957	0.950	0.980
Subway 2	0.659	0.670	0.926	0.931

4.3.2 Approach I vs approach II

To add lower weight to coefficient errors $\delta\boldsymbol{\eta}$ corresponding to less stable dictionary elements, the coefficient error computation scheme (Eq. 3.11) is modified to include a weight vector $\boldsymbol{\Psi}$ (Eq. 3.17). Table 4.2 shows the accuracy in terms of AROC for both approaches (approach

I and approach II, explained at page 3.3). AROC drops a bit on using approach II for Minnesota dataset; however, there is a significant improvement on using approach II in the case of Subway 2 dataset.

In the case of the Minnesota dataset, AROC is already high, which implies the samples do not contain many disturbances. Fig. 4.3 shows the coefficients computed on training samples using $\hat{D}$ for two datasets the Scene 2 and the Subway 2. The figure shows that two dictionary elements (DE) for Scene 2 have similar variance as the two DEs for Subway 2; however, the other 2 DEs for Subway 2 show considerably high variations than the other 2 DEs in Scene 2. Since Approach II gives less weight to unstable DEs, it shows significant improvement in AROC in the case of Subway 2. Table 4.3 shows F1-measures for all dataset, approach II gives higher F1-measure for almost all cases, but the improvement is significantly higher for Subway 2.

4.3.3 Accuracy comparison with other approaches

The accuracy of the proposed algorithm has been compared with the published accuracy of other existing algorithms [16, 43, 61]. All experiments are done with the Minnesota dataset and comparisons are done using AROC. Although the comparison shows that our approach gives a competitive accuracy as compared to the best published accuracy for the Minnesota dataset, to make a reliable comparison is difficult because of the absence of a standard method of the accuracy measurement, as explained below.

There are two ways of counting number of positive and negative events in a panic detection output. Zhao *et al.* [63] takes the whole set of continuous frames corresponding to panic as single event and even if there is one panic detection in that region, it is counted as one true positive and similarly even if there is one false alarm in normal frames then it is counted as one false positive. Another way of computing the number of positive and negative test cases is in terms of the number of frames (Fig. 2.9). Though Mehran *et al.* [43] never mentions explicitly his method of counting test results, from their explanation of results it appears that they have used frame based counting. These two methods will have considerable impact on the reported accuracy. We are using the frame counting based method to report accuracies.

Mehran *et al.* [43] and Wu *et al.* [61] report accuracies for the whole Minnesota dataset and Cong *et al.* [16] report results for each set (Table 4.1) in Minnesota dataset separately. For the sake of consistency, Table 4.4 includes an average of AROC for all samples in Minnesota dataset for the proposed approach and Cong *et al.*'s results. Results of both

Table 4.4: The accuracy comparison for Minnesota dataset (Table 4.1) in terms of AROC. The proposed approach gives a competitive result with respect to the best accuracy on the Minnesota dataset.

Source	**AROC**
Chaotic Invariant [61]	0.99
Social Force [43]	0.96
Optical Flow [43]	0.84
Nearest Neighbour [16]	0.93
Sparse Reconstruction [16]	0.98
Our approach I	0.99
Our approach II	0.98

the proposed approaches (detailed in page 44) are reported using optical flow by Brox *et al.* [11].

A competitive accuracy is obtained using the proposed method, which has simplified existing dictionary learning based method [16] for panic detection. The simplification is achieved by using computationally less expensive OMP and avoiding the use of motion direction. Certainly, in certain anomaly cases, such as direction violation in an escalator, the motion direction is important but panic detection in human crowds may not need this feature.

4.3.4 Time complexity

The proposed algorithm has been tested on a computer with an AMD Athlon(tm) $II \times 3$ 445, 3.10 GHz processor and 4GB RAM. Typically optical flow is computationally expensive and the time complexity of optical flow estimation can vary considerably depending on the implementation. For example, in our computer, the optical flow by Brox *et al.* [11] takes 2.6s and the SIFT flow [38] takes 19s for a 240×320 image. To avoid bias due to the optical flow techniques, the reported time complexity of the proposed algorithm does not include time complexity of optical flow computation. The Matlab implementation of the proposed algorithm runs at 45 frames/s for a video with the frame size 240×320 and it takes 14s to generate training data using 400 samples.

Comparing time complexity with other existing methods is difficult because of differences in the implementation platform and programming language. Moreover, different

algorithms use different optical flow estimation methods and none report their complexity excluding the time complexity of optical flow.

4.3.5 Spatial localization

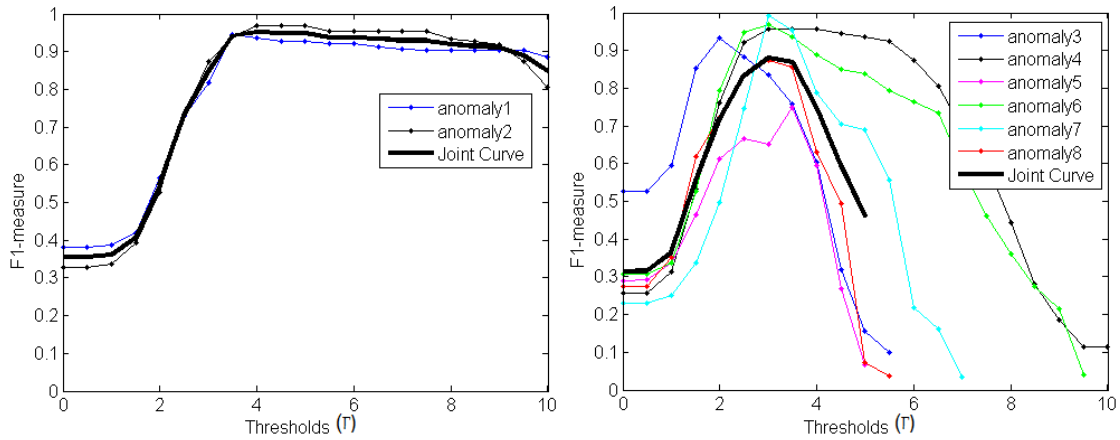
The proposed approach learns $\bar{\mathbf{h}}$ and $\boldsymbol{\sigma_h}$ based on training samples and uses them to spatially localized panic (Sec. 3.4). Fig. 4.6 shows a few test frames with spatial localization.

4.4 Conclusion

SIFT flow is less noisy but a bit pixelated than optical flow. Results show that SIFT flow does not make any improvement in the accuracy of panic detection. The presence of small noise in the optical flow does not make a significant contribution to the HOF for whole image frame. Moreover, as consecutive frames are used for the motion analysis, not much displacement based error is introduced by optical flow to affect accuracies.

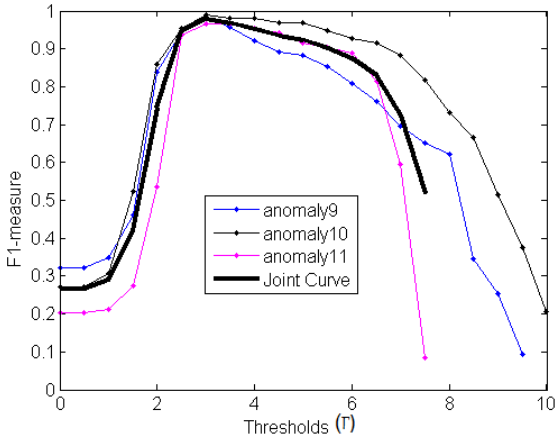
Two variants of panic detection system (listed at page 44) have been proposed. Approach II proposes to use weights to penalize each dictionary element depending on the corresponding stability whereas approach I treats all DEs equally. Approach II does not help much in the Minnesota dataset [1] (Table 4.2); however, it makes significant improvement in Subway 2 dataset [2]. A similar trend is seen in F1-measure. Approach II seems to improve results when a dataset contains many variations in the motion pattern, these variations lead to a set of unstable dictionary elements. These unstable dictionary elements are given less weight in approach II, which improves results.

The accuracy of the proposed method has been compared with other existing methods (Sec. 4.3.3) and the proposed approach gives competitive accuracy with respect to the most accurate approach for the Minnesota dataset (Sec. 4.3.3). A significant accuracy is obtained with respect to the dictionary learning based approach [16]. As there is no standard method to count the number of positive and negative cases (Sec. 4.3.3); it is difficult to make a direct inferece about which one is the best. However, based on the competitive accuracy, and the real-time computational complexity the proposed method should be very effective in real-time panic detection.

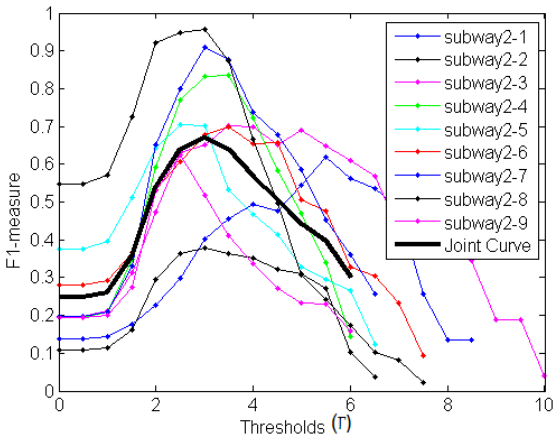


(a) Scene 1

(b) Scene 2



(c) Scene 3



(d) Subway 2

Figure 4.5: Each plot shows F1-measure for a sample set. The joint curve (thick black curves) for a sample set (Table 4.1) represents the average F1-measure of all samples in that set using approach I (page 3.3). Other curves represent F1-measures corresponding to the individual samples. The F1-measure for Scene 2 is a bit lower because of anomaly5 sample, where during the panic people ran along the camera view and after certain distance their speeds were almost equal to the normal speed, giving a false impression of normal behaviour. In the case of Subway 2, the overall accuracy is low because of very low accuracy for sample2-2 and sample2-1. Possible reasons are explained in Sec. 4.1. In (b), (c), and (d) the joint curve stops earlier than a few F1-measure curve because for atleast one sample the F1-measure curve is stopped early to avoid zero F1-measure.

(a) Scene 2 (b) Scene 2, panic localized

(c) Scene 3 (d) Scene 3, panic localized

(e) Subway 2 (f) Subway 2, panic localized

Figure 4.6: Showing three panic containing frames (a), (c), and (e) for which panic is spatially localized (Sec. 3.4) in Fig. (b), (d), and (f). Green filed regions represent panic.

Chapter 5

Conclusions

5.1 Summary

Chapter 4 shows that the proposed OMP based panic detection method produces better accuracy than a state-of-the-art dictionary learning based method [16]. The optical flow estimation is the major contributor to the time complexity of panic detection algorithms (Sec. 4.3.4), it makes a panic detection algorithm non-realtime. Optical flow estimates both the motion direction and the magnitude. The proposed approach uses only the magnitude of motion, we think that inexpensive motion estimation algorithms such as [53] can be developed which provide only motion magnitude.

Table 4.4 compares the accuracy of existing algorithms with the proposed algorithm and it suggests that the proposed panic detection does not compromise on the accuracy because of ignoring motion direction. Accuracies are compared for two types of motion estimation, namely optical flow and SIFT flow (Table 4.2). The SIFT flow is less erroneous, but it is pixelated (Fig. 2.5). The accuracy of panic detection does not look very sensitive to these differences in the flow estimate, it shows that the proposed method is robust against different types of motion estimations.

The proposed approach also provides a simple method to highlight panicked regions in the frame. Sec. 4.3.4 talks about the time complexity of the proposed system, and it shows that if we can have a real-time motion estimation, the proposed algorithm will run in real time. The time complexity could not be compared with existing methods because of the lack of common experimental set-up across different algorithms.

5.2 Future work

One of the simplifications achieved in the proposed work is the use of only motion magnitude. The proposed algorithm uses optical flow [11] to get a motion estimate. Optical flow estimation is computationally expensive (Sec. 4.3.4) and it gives both the magnitude and direction of the motion. A relatively inexpensive method such as [53] can be developed which produces only motion magnitude and tested with the proposed algorithm.

Although the proposed algorithm has not used adaptation, an adaptation mechanism can be useful. Usually panic happens suddenly, if there is a smooth change in the motion behaviour then it should not be called panic. Without adaptation the proposed algorithm will detect even a smooth change as a panic; with adapted dictionary elements and coefficients, the system will take care of smooth transition and it will trigger an alarm only when there is a sudden change.

In the subway dataset low accuracy is observed (Fig. 4.4d) because there were a few cases where people walked perpendicularly to the camera axis (Fig. 4.1b) and there were not enough samples of that type. Such kind of a previously unobserved behaviour can be an anomaly but not a panic situation. A preprocessing step to correct motion magnitude based on the depth can be useful. With the depth information the motion in each position and direction will have the similar estimated value for normal behaviors, and panic will be detected only if there is a change in motion behaviour.

References

[1] Unusual crowd activity dataset made available by the university of minnesota. Downloaded from http://mha.cs.umn.edu/Movies/Crowd-Activity-All.avi.

[2] A. Adam, E. Rivlin, I. Shimshoni, and D. Reinitz. Robust real-time unusual event detection using multiple fixed-location monitors. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 30(3):555–560, March 2008.

[3] M. Aharon, M. Elad, and A. Bruckstein. K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Trans. on Signal Processing*, 54(11):4311–4322, Nov. 2006.

[4] B. Alexe, T. Deselaers, and V. Ferrari. What is an object? *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 73–80, 2010.

[5] S. Ali and M. Shah. A lagrangian particle dynamics approach for crowd flow segmentation and stability analysis. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 1–6, June 2007.

[6] E. L. Andrade, S. Blunsden, and R. B. Fisher. Modelling crowd scenes for event detection. In *18th international conference on Pattern Recognition*, pages 175–178. IEEE, 2006.

[7] A. Basharat, A. Gritai, and M. Shah. Learning object motion patterns for anomaly detection and improved object detection. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008.

[8] S. S. Blackman. Multiple hypothesis tracking for multiple target tracking. *Aerospace and Electronic Systems Magazine, IEEE*, 19:5–18, 2004.

[9] Y. Boers and J. N. Driessen. Multitarget particle filter track before detect application. *IEEE Proc. on Radar, Sonar and Navigation*, 151(6):351– 357, Dec. 2004.

[10] T. Brox, C. Bregler, and J. Malik. Large displacement optical flow. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 41–48, June 2009.

[11] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. High accuracy optical flow estimation based on a theory for warping. In *Computer Vision - ECCV 2004*, volume 3024, pages 25–36, 2004.

[12] P. J. Burt. Fast filter transforms for image processing. *Computer Graphics and Image Processing*, 16(1):20–51, 1981.

[13] A. B Chan and N. Vasconcelos. Mixtures of dynamic textures. In *Tenth IEEE International Conference on Computer Vision*, volume 1, pages 641–647, 2005.

[14] M. T. Chan, A. Hoogs, J. Schmiederer, and M. Petersen. Detecting rare events in video using semantic primitives with hmm. In *Proc. of the 17th international conference on Pattern Recognition*, volume 4, pages 150–154, 2004.

[15] V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: A survey. *ACM Computing Surveys*, 41(3):1–58, 2009.

[16] Y. Cong, J. Yuan, and L. Liu. Sparse reconstruction cost for abnormal event detection. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 3449–3456, June 2011.

[17] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, pages 886–893, Los Alamitos, CA, USA, 2005.

[18] H. Dee and D. Hogg. Detecting inexplicable behaviour. In *British Machine Vision Conference*, volume 477, pages 477–486. Citeseer, 2004.

[19] J. Diehl, F. Gathmann, F. Hans, and J. Juttner. The facts behind the duisburg disaster. http://tinyurl.com/ce68zjs, July 2010.

[20] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*, volume 2. Wiley, 2001.

[21] C. Fei, R. H. Kwong, and D. Kundur. A hypothesis testing approach to semifragile watermark-based authentication. *IEEE Trans. on Information Forensics and Security*, 4(2):179–192, June 2009.

[22] J. J Fruin. The causes and prevention of crowd disasters. *Science*, pages 1–10, 1993.

[23] D. Gao and N. Vasconcelos. Decision-theoretic saliency: computational principles, biological plausibility, and implications for neurophysiology and psychophysics. *Neural Computation*, 21(1):239–271, 2009.

[24] P. Grassberger and I. Procaccia. Characterization of strange attractors. *Physical Review Letters*, 50(5):346–349, 1983.

[25] C. Harris. The privacy balance. http://www.claimscanada.ca/issues/article.aspx?aid=1000217415.

[26] D. Helbing and P. Molnar. Social force model for pedestrian dynamics. *Physical Review E*, 51(5):4282–4286, 1998.

[27] B. K. P. Horn and B. G. Schunck. Determining optical flow. *Articial Intelligence*, 1980.

[28] W. Hu, X. Xiao, Z. Fu, D. Xie, T. Tan, and S. Maybank. A system for learning statistical motion patterns. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 28(9):1450–1464, Sept. 2006.

[29] K. Huang and S. Aviyente. Sparse representation for signal classification. *Neural Information Processing Systems*, 19(3):609–616, 2007.

[30] P. Indyk and R. Motwani. Approximate nearest neighbors: towards removing the curse of dimensionality. In *Proc. of the 30th annual ACM symposium on Theory of computing*, STOC '98, pages 604–613, 1998.

[31] N. Johnson and D. Hogg. Learning the distribution of object trajectories for event recognition. *Image and Vision Computing*, 14(8):609–615, 1996.

[32] L. Kratz and K. Nishino. Anomaly detection in extremely crowded scenes using spatio-temporal motion pattern models. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 1446–1453, June 2009.

[33] S. Kullback and R. A. Leibler. On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86, 1951.

[34] A. Kumar, F. Tung, A. Wong, and D. A. Clausi. A decoupled approach to illumination-robust optical flow estimation. *Under review*, 2012.

[35] A. Kumar, A. Wong, A. Mishra, D. A. Clausi, and P. Fieguth. Tensor vector field based active contours. In *IEEE International Conference on Image Processing*, 2011.

[36] M. P. Kumar, P. Torr, and A. Zisserman. Learning layered motion segmentations of video. *International Journal of Computer Vision*, 76(3):301–319, 2008.

[37] W. Li and X. Wu. Crowd density estimation: An improved approach. In *IEEE 10th international conference on Signal Processing*, pages 1213–1216, Oct. 2010.

[38] C. Liu, J. Yuen, A. Torralba, J. Sivic, and W. T. Freeman. Sift flow: Dense correspondence across different scenes. In *Proc. of the 10th European Conference on Computer Vision Part III*, volume 1, pages 28–42. Springer, 2008.

[39] David G. Lowe. Object recognition from local scale-invariant features. In *IEEE International Conference on Computer Vision*, volume 2, page 1150, Los Alamitos, CA, USA, 1999.

[40] V. Mahadevan, W. Li, V. Bhalodia, and N. Vasconcelos. Anomaly detection in crowded scenes. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 1975–1981, June 2010.

[41] S. G. Mallat and Z. Zhang. Matching pursuits with time-frequency dictionaries. *IEEE Trans. on Signal Processing*, 41(12):3397–3415, Dec. 1993.

[42] J. S. Marques, P. M. Jorge, A. J. Abrantes, and J. M. Lemos. Tracking groups of pedestrians in video sequences. In *Conf. on Computer Vision and Pattern Recognition Workshop, CVPRW '03*, volume 9, page 101, June 2003.

[43] R. Mehran, A. Oyama, and M. Shah. Abnormal crowd behavior detection using social force model. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 935–942, June 2009.

[44] A. Mittal and N. Paragios. Motion-based background subtraction using adaptive kernel density estimation. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, volume 2, pages 302–309, 2004.

[45] M. Mucientes and W. Burgard. Multiple hypothesis tracking of clusters of people. In *IEEE international conference on Intelligent Robots and Systems*, pages 692–697, 2006.

[46] H. T. Nguyen, Qiang J., and A. W. M. Smeulders. Spatio-temporal context for robust multitarget tracking. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 29(1):52–64, 2007.

[47] Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad. Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition. In *27th Conf. on Signals, Systems and Computers*, volume 1, pages 40–44, Nov. 1993.

[48] M. Patzold, R. H. Evangelio, and T. Sikora. Counting people in crowded environments by fusion of shape and motion information. In *7th IEEE international conference on Advanced Video and Signal Based Surveillance*, pages 157–164, 2010.

[49] M. Piccardi. Background subtraction techniques: a review. In *IEEE international conference on Systems Man and Cybernetics*, volume 4, pages 3099–3104. IEEE, 2004.

[50] H. Qian, Y. Mao, W. Xiang, and Z. Wang. Recognition of human activities using svm multi-class classifier. *Pattern Recognition Letter*, 31(2):100–111, Jan. 2010.

[51] V. Reddy, C. Sanderson, and B. C. Lovell. Improved anomaly detection in crowded scenes via cell-based analysis of foreground speed, size and texture. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2011.

[52] D. Reynolds. Gaussian mixture models. *Tech. report, MIT Lincoln Laboratory*, 45(2):1–5, 2008.

[53] A. H. Shabani, D. A. Clausi, and J. S. Zelek. Improved spatio-temporal salient feature detection for action recognition. In *Procedings of the British Machine Vision Conference 2011*, volume 1, 2011.

[54] D. Sun, S. Roth, and M. J. Black. Secrets of optical flow estimation and their principles. In *IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, pages 2432–2439, Los Alamitos, CA, USA, 2010.

[55] R. Szewczyk, K. Grabowski, M. Napieralska, W. Sankowski, M. Zubert, and A. Napieralski. A reliable iris recognition algorithm based on reverse biorthogonal wavelet transform. *Pattern Recognition Letters*, 33(8):1019–1026, 2012.

[56] P. Tu, T. Sebastian, G. Doretto, N. Krahnstoever, J. Rittscher, and T. Yu. Unified crowd segmentation. In *Proc. of European Conference on Computer Vision*, chapter 51, pages 691–704. 2008.

[57] Langdon. W. Receiver operating characteristics (roc). http://http://www.cs.ucl.ac.uk/staff/ucacbbl/roc/.

[58] S. Wang and Z. Miao. Anomaly detection in crowd scene using historical information. In *International Symposium on Intelligent Signal Processing and Communication Systems*, pages 1–4, Dec. 2010.

[59] A. Wedel, D. Cremers, T. Pock, and H. Bischof. Structure- and motion-adaptive regularization for high accuracy optic flow. In *IEEE 12th international conference on Computer Vision*, pages 1663–1668, Oct. 2009.

[60] A. Wolf, J. B. Swift, H. L. Swinney, and J. A Vastano. Determining lyapunov exponents from a time series. *Physica D: Nonlinear Phenomena*, 16(3):285–317, 1985.

[61] S. Wu, B. E. Moore, and M. Shah. Chaotic invariants of lagrangian particle trajectories for anomaly detection in crowded scenes. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 2054–2060, 2010.

[62] B. Zhan, D. N. Monekosso, P. Remagnino, S. A. Velastin, and L. Q. Xu. Crowd analysis: a survey. *Machine Vision and Applications*, 19(5-6):345–357, 2008.

[63] b. Zhao, L. Fei, and E. P. Xing. Online detection of unusual events in videos via dynamic sparse coding. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 3313–3320, June 2011.