Saliency Guided Video Anomaly Detection

Submitted in partial fulfillment of the requirements
of the degree of
Master of Electronics & Telecommunication Engineering

by

Ranodip Das
Registration No. 128924 of 2014-2015
Examination Roll No. M4ETC1609

Supervisor
Dr. Ananda Shankar Chowdhury
DEPARTMENT OF ELECTRONICS & TELECOMMUNICATION ENGINEERING
JADAVPUR UNIVERSITY - KOLKATA

May, 2016

Faculty of Engineering & Technology

Jadavpur University

CERTIFICATE

This to certify that the thesis entitled "**Saliency Guided Video Anomaly Detection**" has been carried out by Ranodip Das (Registration No. 128924 of 2014 - 15) under my guidance and supervision and be accepted in partial fulfillment of the requirement for the degree of Master of Engineering in Electronics and Telecommunication Engineering.

Supervisor

Dr. Ananda Shankar Chowdhury
Associate Professor
Department of Electronics andTele-communication Engineering
Jadavpur University
Kolkata-700032

Head of the Department

Dr. Palaniandavar Venkateswaran
Professor
Department of Electronics andTele-communication Engineering
Jadavpur University
Kolkata-700032

Prof. Sivaji Bandyopadhyay
Dean, Faculty of Engineering and Technology
Jadavpur University
Kolkata – 700032

FACULTY OF ENGINEERING AND TECHNOLOGY
JADAVPUR UNIVERSITY

Thesis Approval*

The thesis entitled

Saliency Guided Video Anomaly Detection

by

Ranodip Das

Registration No. 128924 of 2014-15

Examination Roll No. M4ETC1609

is approved for the degree of

Master of Engineering in Electronics & Telecommunication Engineering

---------- ----------

Signature of the Examiner Signature of the Supervisor

Date: _____

Place: _____

* Only in the case the thesis is approved

Declaration

I hereby declare that the thesis contains literature survey and research work in my own words and is truly in accordance with the academic ethics andeducational integrity. With my sole responsibility I would like to mention that none of the ideas, facts, results represented in the thesis have been fabricated or falsified. I have cited and referenced the original sources as and when required. I understand that non-compliance of any of the

---------- **Date……..**

Ranodip Das

Reg. No.: 128924 of 2014-15

Acknowledgements

I am grateful to my supervisor Dr.Ananda Shankar Chowdhury for the exclusive training and motivation he hasoffered me during the entire course of my thesis work. His unique guidance andsupport made it possible for me to work on a topic that was of great interest to me. His optimistic attitude towards achieving a difficult goal and the crucial ideas that hehas come up with every now and then has boosted my belief and confidence to complete the project. I will carry the fire ofambition that he has ignited in me while transcending to the next level.I am profoundly grateful to Professor Palaniandavar Venkateswaran, Head of the Department,Electronics and Telecommunication Engineering Department, Jadavpur University who was kind enough to provide me with all the necessary facilities to carry out this project.

I thank all the members of our group "Imaging, Vision and Pattern Recognition (IVPR)". Working in this group has been a wonderful and memorable experience to me. I want to extend gratitude to my seniors Vijay N. Gangapure and Susmit Nanda who havehelped me at various critical junctures during my research. My co-worker of the Control Engineering Laboratory, Jadavpur University, Ms.Rukhmini Roydeserves specialmention for keeping the environment of the laboratory suitable for research work and also guiding me at times. I also thank the Control Laboratory office staff at Jadavpur University, Shyamal C. Laha for his support at different times.

I am indebted to a number of my friends Mainak Dan, Srishti Srivastava, Ranita Saha and Rimita Lahiri for providing a stimulating and fun filled atmosphere and disbursal unforgettable moments in Jadavpur University.I might likewise want to express gratitude toward some of my close friends Santipriya Singh, Avishek Sarkar, Sai Krishna, Jagadish Mahato and Deep Kayal who have constantly bolstered me and roused me toward higher studies.

At long last and above all, I owe an extraordinary obligation of appreciation to my parents Mr. Ranjit Kr. Das and Mrs. Nabanita Das who have dependably been there close by to serve every one of the necessities of my existence with extraordinary consideration and who have taken away all the outside weights to give me the space to buckle down. My sister Ms. Ranita Das has constantly been a strong motivation for me as she always finds an optimisticelement in any problem.Last however not the minimum, I am appreciative to my special friendMs. Dipannita Karmakar who has supported me through out and made me believe in myself. The fulfillment from the achievement of this work would stay fragmented without saying thanks to her cordially.

Ranodip Das

Abstract

The problem of video anomaly detection has drawn lots of attention in the recent past for the researchers in both the computer vision and the multimedia communities. By an anomalous region in an image or video, we mean a region with some suspicious event or action. There have been many methods proposed to detect an anomalous activity which are discussed in this thesis briefly. In this thesis work, the concept of saliency has been used as a cue in the model proposed to detect a video anomaly. Saliency, as it means is something that puts itself into attention and naturally human gaze is easilymore concentrated on a visually salient region. Saliency detection techniques can be used fordiverse applications e.g. anomaly detection, suspicious activity detection, object shape detectionand in general for modeling of human gaze. So the basic idea of the proposed work is based on the fact that an anomalous event or activity will be salient or highlighted than other normal activities going around in the same scenario, and will thus catch human attention. This thesis work uses motion context to detect an anomaly. Further, the motion cue is extracted from two different approaches, one from pixel level optical flow and the other from superpixel level saliency map. The two motion cues are then fused together which gives a combined motion description of each individual. 3D-DCT is used for object association between frames and stable individuals, called *observers* are stored. The motion variation of these observers with its neighbors across frames is observed and if the value is greater than a certain threshold, the frame is declared as an anomalous frame.

Dedicated to my Family and all my countrymen

Contents

CHAPTER 4: EXPERIMENTAL RESULTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

Introduction

This chapter provides a synopsis of the thesis. In section 1.1, the problem of video anomaly detection is introduced, where first an outline of video anomaly is put in followed by a brief of the problem. Section 1.2 gives an overview of video saliency and its utility in detecting anomalies. The motivation behind the thesis is listed in Section 1.3. Section 1.4 presents the key contributions of the thesis. Finally, the chapter is concluded with an overview of the organization of the thesis in Section 1.5.

1.1 Video Anomaly Detection

At present,the World is drowning in the deluge of data that are being collected, but also at the same time we are starving for information and knowledge from them. However the data collected are not always normal; there may be some data that can lead to crucial patterns or activities. The study of these patterns is more comfortable if the data obtained is in the form of video.

1.1.1 What is Anomaly?

Anomaly is a pattern in the data that does not correspond to the expected behavior.It is also referred to as outliers, exceptions, peculiarities, surprise, etc. [2]. Anomalous events occur relatively infrequently.Nevertheless when they do occur their consequences can be quite dramatic and mostly in a negative sense. Anomaly detection is an important problem that has been researched within diverse research areas and application domains [16], [21]-[24]. Many anomaly detection techniques are application specific whereas many of them are more generic.

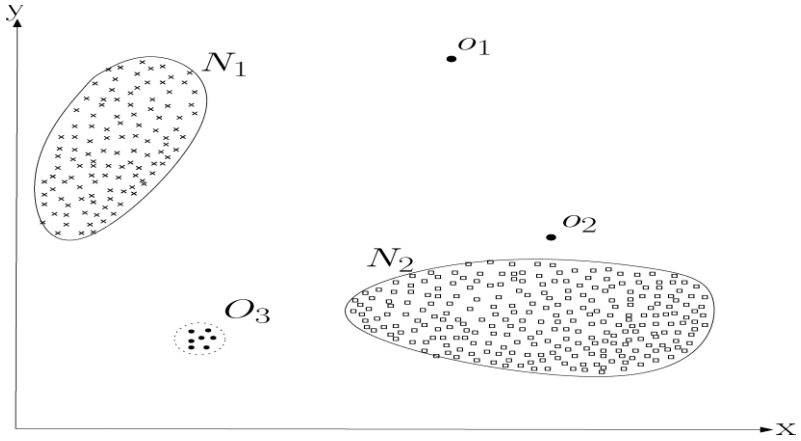


Figure 1: A simple example of anomalies in a two-dimensional data set.

Figure 1 illustrates anomalies in a simple two-dimensional data set. The data has two normal regions, *N*1 and *N*2, since most observations lie in these two regions. Points that are sufficiently far away from these regions, for example, points *o*1 and *o*2, and points in region *O*3, are anomalies [1].

An aboveboard anomaly detection approach is to define a region which represents a normal behavior and then adjudge any data as anomaly that does not correspond to the normal region. But there are many factors that impede this simple approach of anomaly detection technique:

- A perfect normal region cannot be defined that comprehends every possible normal behavior. Also the boundary between normal and anomalous region may not be precise and hence detection of the observations near the boundary becomes difficult.

- A major issue is the availability of the labeled data for training of models used by the anomaly detection techniques.
- The data may sometimes contain noise and hence normal data added with noise may be mistaken as anomalous data or even anomalous observations may appear normal, thus making it difficult to define a normal behavior and also to detect an anomaly.
- In many fields the normal behavior keeps developing / evolving and hence a current definition of normal behavior might not be sufficient to illustrate the future normal behavior.
- The exact definition of anomaly varies for different application domains. A small deviation in medical domain might be considered as an anomaly, whereas the same fluctuation might be considered as normal in the stock market domain. Thus a technique developed for anomaly detection in one domain cannot be applied on another domain.

Thus, the anomaly detection problem is not easy to solve in its most general form. Most of the existing techniques solve a particular specification of the problem. The various factors on which the formulation of the problem depends are nature of the data, availability of labeled data, type of anomaly to be detected and so on. These factors are determined by the application domain in which the anomaly is to be detected. Concepts from many diverse disciplines such as statistics, machine learning, data mining, information theory, spectral theory have been adopted and applied them to specific

problem formulations. The key components associated with an anomaly detection technique have been shown in Figure 2.

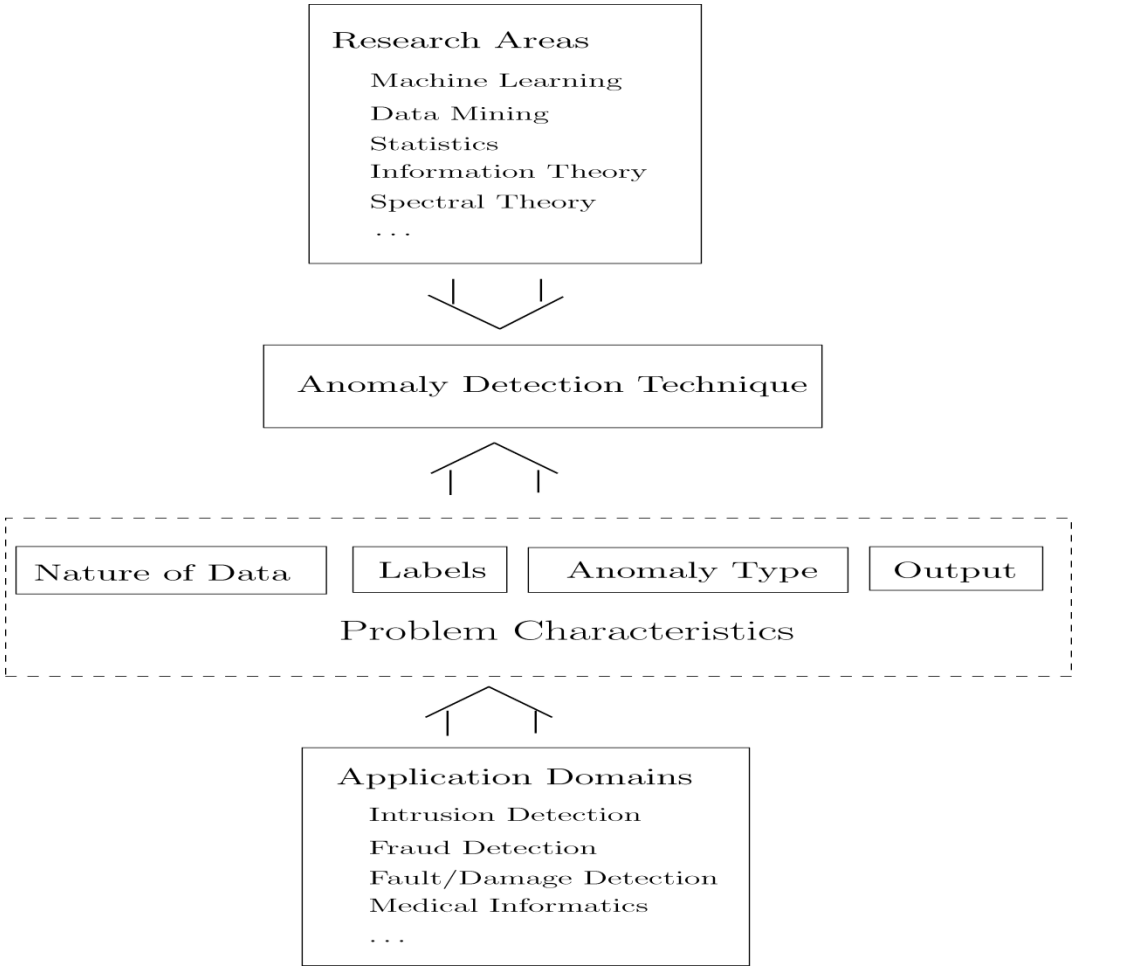


Figure 2: Key components of an anomaly detection technique.

1.1.2 Video Anomaly Detection

Video has always been a better and enriched source of information. We all are aware of the fact that the criminal activities are increasing all over the world, and thus the

CCTV video footagesare of great importancein places like ATMs, busy roads, shops, railway stations, airports, research centers etc. where security is a major issue. The information obtained from a surveillance video is useful in detecting the suspicious activity or events and may sometimes even detect a suspicious person. However, manual detection of such activities from those videos is a cumbersome task. Hence a model is necessary which can automatically detect an anomaly and points out the instant at which the suspicious activity has occurred.Therequirementof developing such a model has made the surveillance domain as one of the most researched fields in computer vision.

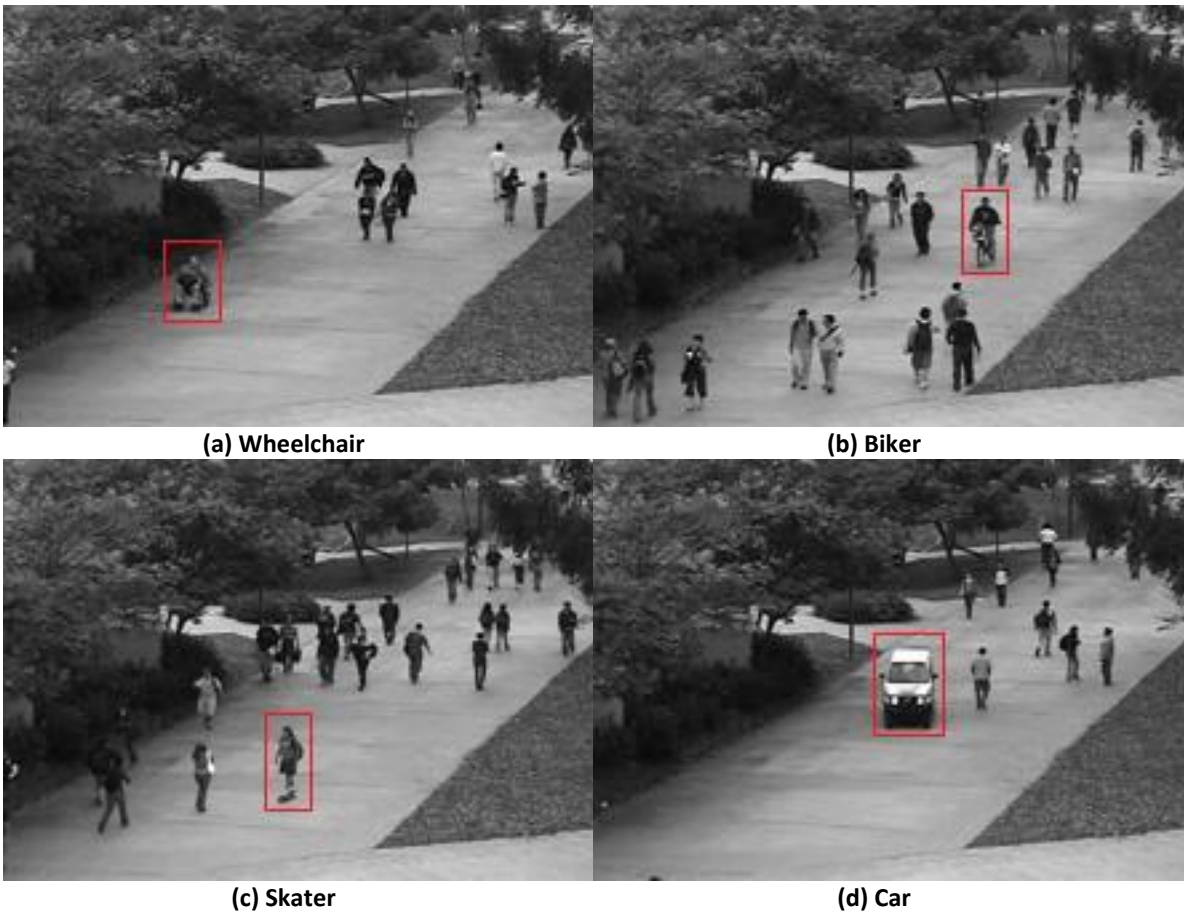


Figure 3: Examples of anomalous activities shown in four different cases where (a)Wheelchair (b)Biker (c) Skater (d) Car are seen in the walkways.

The main aspect to look for while solving video anomaly detection [25]-[28] problem is the motion context or modeling the motion patterns. The main problem is to detect the anomalous frames of a video with high density crowd scenes using the motion context of stable individuals (called *observers*) obtained from the temporal saliency map.

To solve this problem, one may have to complete the following major objectives:

a) The first task is to detect all the pedestrians in a given frame.
b) Secondly, from the temporal saliency map the saliency weight of each individual is to be obtained. Variations of these weights of the observers with its neighborhoodsmust be studied between two consecutive frames.If the variation is above a certain threshold, the particular frame should be declared as an anomalous frame.

1.2 Video Saliency

A video can be looked upon as many still images which are mutually correlated to each other. Such consecutive images shown with a definite frame rate would appear as a video to an observer. The human visual system can process 10 to 12 separate images per second as distinct individual images, and sequences at higher rates are perceived as motion [3].Usually, the human visual system does not look at every object in a visual scene but it automatically seeks prominent regions and movements to reduce search efforts [4]. Thus the salient regions and the salient actions in videos attract human attention more than the non-prominent regions.

Saliency may be defined on the grounds of region dependent features like color, texture and luminance or dynamic features like motion, velocity corresponding to both magnitude and direction. Generally, the modeling of saliency for videos includes both the static and dynamic features. The basic doctrine of detection in most of the saliency models is to extract the static features first and then use them to extract the dynamic features. The final saliency detection is obtained by combining the extracted static and dynamic features. The consequence of video saliency is not just to make a decision about an object to be salient or non-salient, but it is equally important to set a membership/ score of saliency of the objects, i.e. the detected salient objects have a ranking/ score/ weight based on saliency. This marking of the ranks of the salient objects completes the model of saliency. The human brain not only determines a salient object in a visual scene but also assigns a rank of the saliency. To model this ranking scheme of salient objects in an algorithm, a stochastic model for visually salient regions is required [5]. Such a model will depend on various factors affecting the human visual process. Finally, an optimization process is required that integrate these factors and then a logical combination strategy combines them all. Thus the problem of saliency detection can be thought of as a problem of multi-criteria optimization problem.

1.3 Motivation

The concept of anomaly detection has wide range of applications in the field of image processing. The anomaly detection techniques that deal with images are either concerned with any changes in an image over time, i.e. motion detection or in regions

that appear abnormal on the static image. Many specific problems have been researched in this domain that includes satellite imagery [6]-[10], digit recognition [11], spectroscopy [12]-[15], mammographic image analysis [16]-[17], and video surveillance [18]-[20]. The anomalies are caused due to motion, or insertion of a foreign object, or instrumentation errors. The data has spatial as well as temporalfeatures. Each data point has a few continuous attributes such as color, lightness, texture, and luminance and so on.The concerning anomalies are either anomalous points corresponding to point anomalies or particular regions in the images corresponding to contextual anomalies.

One of the key challenges in this domain is the large size of the input data. Online anomaly detection techniques are required while dealing with video data that can detect anomalies in real-time scenarios. Some anomaly detection techniques used in this domain are regression [10], [12], Bayesian networks [20], support vector machines [13], neural networks [6], [8], [11], [14], [18], mixture of models [7], [16], [17], Clustering [15], nearest neighbor-based techniques [7], [19].

Although anomaly detection has attracted the attention of researchers in the computer vision and multimedia fields for quite a long time, most of the research works have been conducted on still images, taking care of only the features of static nature viz. color, texture, orientation and luminance or on videos where the crowd density is relatively low. Abnormal behavior detection in the high density crowd scenes in a video is always a challenging problem in the field of computer vision. There has been very few research works done on videos with high density crowd scenes.

The manual detection of any anomalous event or activity in a surveillance video footage is a tedious and cumbersome job. Also most of the surveillance videos require the anomaly to be detected at the instant when it occurred. Thus designing a model which can directly detect the anomalous event or suspicious activity in no time is necessary. This is the main motivation that led me to work in this type of problem.

1.4 Key Contribution to the Thesis

The motion context of each individual is extracted from the temporal saliency map obtained from a superpixel levelsaliency model. This further helps in developing the relational motion pattern between any two individuals called as connecting weight. The motion cue thus obtained is fused with the motion information extracted from pixel level optical flow model. This fused weight describes the motion context between any two individuals in a frame, which acts as the primary information in anomaly detection.

1.5 Organization of the Thesis

The rest of the thesis is organized into following four chapters: **Chapter 2**provides thedetailed description of ‘saliency’. The chapter includes description of the term ‘saliency’ in the field of computer vision. This chapter also illustrates video saliency briefly and finally ends with a detailed description of the saliency map.**Chapter 3**covers the description of the proposed solution with details of each block that constitutes the proposed anomaly detection model. This chapter provides the necessary theoretical background of each block, its mathematical implementation and its usage in the thesis

work. **Chapter 4** illustrates the dataset on which the model is implemented, the parameter settings, computing platform and finally the results obtained after each block. The chapter ends with a comparison of the proposed model with some previous methods. **Chapter 5**concludes the thesis indicating some directions of future research.

References

[1] Chandola, Varun, Arindam Banerjee, and Vipin Kumar. "Anomaly detection: A survey." *ACM computing surveys (CSUR)* 41, no. 3 (2009): 15.

[2] Numenta White Paper: The science of Anomaly Detection.

[3] Read, Paul, and Mark-Paul Meyer. *Restoration of motion picture film*. Butterworth-Heinemann, 2000.

[4] Marat, Sophie, Tien Ho Phuoc, Lionel Granjon, Nathalie Guyader, Denis Pellerin, and Anne Guérin-Dugué. "Modelling spatio-temporal saliency to predict gaze direction for short videos." *International journal of computer vision* 82, no. 3 (2009): 231-243.

[5] Avraham, Tamar, and Michael Lindenbaum. "Esaliency (extended saliency): Meaningful attention using stochastic image modeling." *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 32, no. 4 (2010): 693-708.

[6] Augusteijn, M. F., and B. A. Folkert. "Neural network classification and novelty detection." *International Journal of Remote Sensing* 23, no. 14 (2002): 2891-2902.

[7] Byers, Simon, and Adrian E. Raftery. "Nearest-neighbor clutter removal for estimating features in spatial point processes." *Journal of the American Statistical Association* 93, no. 442 (1998): 577-584.

[8] Moya, Mary M., Mark W. Koch, and Larry D. Hostetler. *One-class classifier networks for target recognition applications*. No. SAND-93-0084C. Sandia National Labs., Albuquerque, NM (United States), 1993.

[9] Theiler, James P., and D. Michael Cai. "Resampling approach for anomaly detection in multispectral images." In *AeroSense 2003*, pp. 230-240. International Society for Optics and Photonics, 2003.

[10] Torr, Philip HS, and David W. Murray. "Outlier detection and motion segmentation." In *Optical Tools for Manufacturing and Advanced Automation*, pp. 432-443. International Society for Optics and Photonics, 1993.

[11] Le Cun, B. Boser, John S. Denker, D. Henderson, Richard E. Howard, W. Hubbard,

and Lawrence D. Jackel. "Handwritten digit recognition with a back-propagation network." In *Advances in neural information processing systems*. 1990.

[12] Chen, Da, Xueguang Shao, Bin Hu, and Qingde Su. "Simultaneous wavelength selection and outlier detection in multivariate regression of near-infrared spectra." *Analytical Sciences* 21, no. 2 (2005): 161-166.

[13] Davy, Manuel, and Simon Godsill. "Detection of abrupt spectral changes using support vector machines. an application to audio signal segmentation." In *ICASSP*, vol. 2, pp. 1313-1316. 2002.

[14] Hazel, Geoffrey G. "Multivariate Gaussian MRF for multispectral scene segmentation and anomaly detection." *Geoscience and Remote Sensing, IEEE Transactions on* 38, no. 3 (2000): 1199-1211.

[15] Scarth, G., M. McIntyre, B. Wowk, and R. Somorjai. "Detection of novelty in functional images using fuzzy clustering." In *Proceedings of the 3rd Meeting of International Society for Magnetic Resonance in Medicine*. 1995.

[16] Spence, Clay, Lucas Parra, and Paul Sajda. "Detection, synthesis and compression in mammographic image analysis with a hierarchical image probability model." In *Mathematical Methods in Biomedical Image Analysis, 2001. MMBIA 2001. IEEE Workshop on*, pp. 3-10. IEEE, 2001.

[17] Tarassenko, Lionel, P. Hayton, N. Cerneaz, and M. Brady. "Novelty detection for the identification of masses in mammograms." In *Artificial Neural Networks, 1995., Fourth International Conference on*, pp. 442-447. IET, 1995.

[18] Singh, Sameer, and Markos Markou. "An approach to novelty detection applied to the classification of image regions." *Knowledge and Data Engineering, IEEE Transactions on* 16, no. 4 (2004): 396-407.

[19] Pokrajac, Dragoljub, Aleksandar Lazarevic, and Longin Jan Latecki. "Incremental local outlier detection for data streams." In *Computational Intelligence and Data Mining, 2007. CIDM 2007. IEEE Symposium on*, pp. 504-515. IEEE, 2007.

[20] Dieh, Christopher P., and John B. Hampshire. "Real-time object classification and novelty detection for collaborative video surveillance." In*Neural Networks, 2002.*

IJCNN'02. Proceedings of the 2002 International Joint Conference on, vol. 3, pp. 2620-2625. IEEE, 2002.

[21] Kumar, Vipin. "Parallel and distributed computing for cybersecurity." *IEEE Distributed Systems Online* 10 (2005): 1.

[22] Fujimaki, Ryohei, Takehisa Yairi, and Kazuo Machida. "An approach to spacecraft anomaly detection problem using kernel feature space." In*Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pp. 401-410. ACM, 2005.

[23] Edgeworth, F. Y. "Xli. on discordant observations." *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 23, no. 143 (1887): 364-375.

[24] Aleskerov, Emin, Bernd Freisleben, and Bharat Rao. "Cardwatch: A neural network based database mining system for credit card fraud detection." In*Computational Intelligence for Financial Engineering (CIFEr), 1997., Proceedings of the IEEE/IAFE 1997*, pp. 220-226. IEEE, 1997.

[25] Yuan, Yuan, Jianwu Fang, and Qi Wang. "Online anomaly detection in crowd scenes via structure analysis." *Cybernetics, IEEE Transactions on* 45, no. 3 (2015): 548-561.

[26] Cheng, Kai-Wen, Yie-Tarng Chen, and Wen-Hsien Fang. "Video anomaly detection and localization using hierarchical feature representation and Gaussian process regression." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2909-2917. 2015.

[27] Li, Weixin, Vijay Mahadevan, and Nuno Vasconcelos. "Anomaly detection and localization in crowded scenes." *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 36, no. 1 (2014): 18-32.

[28] Wu, Si, Hau-San Wong, and Zhiwen Yu. "A bayesian model for crowd escape behavior detection." *Circuits and Systems for Video Technology, IEEE Transactions on* 24, no. 1 (2014): 85-98.

CHAPTER 2

Video Saliency

This chapter provides an overview of video saliency which helps as a cue in the detection of video anomaly. Section 2.1 illustrates the basic of saliency. Usefulness of saliency in the field of computer vision is explained in Section 2.2. Section 2.3 deals with video saliency and finally the saliency map is illustrated in Section 2.4 where two approaches of saliency are introduced.

2.1 What is Saliency?

The term salience or saliency of any item refers to having a quality that thrusts itself into attention. Saliency typically arises from contrasts between items and their neighborhood, such as a pedestrian walking along the wrong side of the road or a vehicle moving in a no entry zone. Saliency detection is considered to be a key attentional mechanism that facilitates learning and survival by enabling organisms to focus their limited perceptual and cognitive resources on the most pertinent subset of the available sensory data. The part which is salient can be influenced by training, for example, particular letters can become salient for human subjects by training [5]-[6].

The hippocampus participates in the assessment of salience and context using past memories to filter new incoming stimuli; placing those that are most important into long term memory. The entorhinal cortex is the pathway into and out of the hippocampus.

The term "saliency" is broadly used in the study of perception and cognition to refer to any aspect of a stimulus that stands out from the rest. Salience may be the result of emotional, motivational or cognitive factors and is not necessarily associated with physical factors such as intensity, clarity or size.

2.2 Computer Vision and Saliency

Computer vision is a field that includes methods for acquiring, processing, analyzing, and understanding images and, in general, high-dimensional data from the real world in order to produce numerical or symbolic information, e.g., in the forms of

decisions [8]-[10]. The basic problem in most of the computer vision tasks can be defined as extraction of "significant" descriptions from images or image sequences. The Human Visual System (HVS) uses a combination of image driven data and certain prior models in its processing. Visual saliency is a broad term that refers to the idea that certain parts of a scene are distinctive. It is shown in Figure 4 the different part(s) that are considered as visually salient.

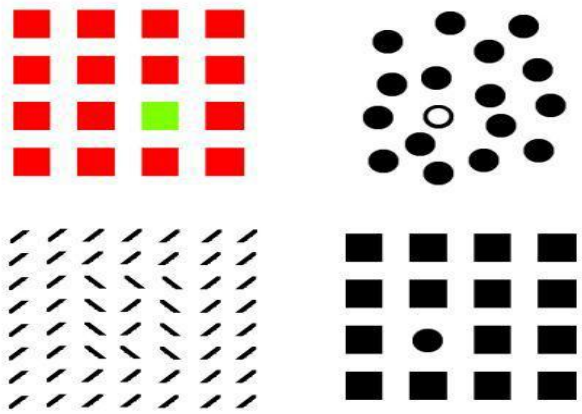


Figure 4: Salient parts shown in four different pictures.

In the first picture of the Figure 4, the green square is distinct visually among all the red squares. The white circle in the second picture among other black circles, differently oriented marks in the third picture and the circular dot among other rectangular boxes in the fourth picture are visually distinct. All of these pictures are immediately drawing attraction.

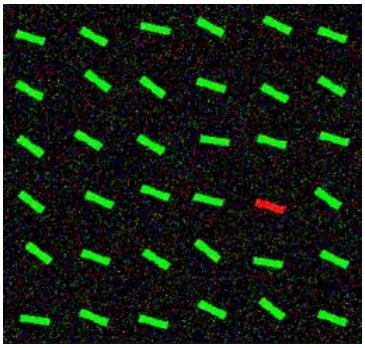
2.3 What is Video Saliency?

Video is defined as recording, reproducing, or broadcasting of moving visual images. Video has consistently been the preferred medium for most people as it gives information in both audio and visual mode thus enriching the content delivery. As the field of digital multimedia is continuously advancing a lot of information is widely available in digital form such as television shows and movies. The advancement of the digital domain has led to the development of network bandwidth and cloud technology.So, the use of digital video recorder such as camcorders, digital cameras and smart phones has multiplied which in turn has added to the availability of enriched video information. This advancement has also helped to accelerate the pace of research in the field of video processing in several angles.

Our attention is attracted to visually salient stimuli. It is important for a biological system to quickly detect potential prey, predators, or mates in a littered visual world. However, due to the computational complexity it is a daunting task to simultaneously identify any and all interesting targets in one's visual field even by best of the biological brains [1]. One basic solution adopted is to limit the complex object recognition process to a smaller area or to limit the recognition to few objects at any one time. Thus many objects or areas in the visual scene can then be processed one after the other. This process of visual scene operation was inspired through mechanisms of visual attention, where a common but somewhat inaccurate metaphor for attention is that of a virtual spotlight, switching to and spotlighting different sub-regions of the visual world, so that one region at a time can be subjected to more detailed and accurate visual analysis [2]-[4].The

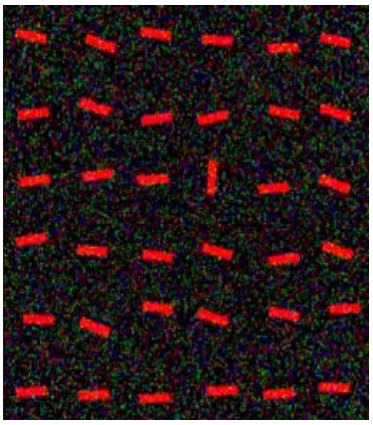
inability to fully process all the regions in parallel might be solved by visual attention. But, this too may produce a problem as of which sub-region or target of attention to be selected. Visual saliency helps the brain to achieve a reasonably efficient selection. Human brain has evolved to quickly compute saliency in an automatic way and in real-time over the entire visual field. Visual attention is then pulled towards salient visual locations.

Visual saliency is sometimes cavalierly described as a physical property of a visual stimulus. Saliency is the consequence of an interaction of a stimulus with other stimuli, as well as with a visual system, whether biological or artificial. As a straight-forward example, a color-blind person will have a dramatically different experience of visual saliency than a person with normal color vision, even when both look at exactly the same physical scene as shown in Figure 5 below.



One bar in the array of bars strongly *pops-out* and immediately attracts attention effortlessly. Many studies have suggested that in simple displays like this, no scanning occurs: Attention is immediately drawn to the salient item, no matter how many other items called *distractors* are present in the display [2], [7]. This suggests that the image is processed in parallel (all at once) to determine saliency at every location and to orient towards the most salient location.

Figure 5: Video Saliency Example 1.



The vertical bar is visually salient. As compared to the above example, local visual properties of a given item do not determine how perceptually salient the item will be; rather, looking at a given item within its surrounding context is crucial. Comparing the red bar in the top-left corner of this image to the salient bar in the image above: both bars are red and both have similar local appearances. Yet the one in the top-left corner here has low saliency and attention is much more strongly attracted to the more salient vertical bar, while the red bar in the above image is highly salient.

Figure 6: Video Saliency Example 2.

2.4 The Saliency Map

The groundwork of saliency map can be found from Feature Integration Theory [2]. It consists of the following elements (Figure 7):

(i) A representation composed of a set of feature maps, computed in parallel, permitting separate representations of several stimulus characteristics.

(ii) A topographic saliency map where each location encodes the combination of properties across all feature maps as a conspicuity measure.

(iii) A selective mapping into a central non-topographic representation, through the topographic saliency map, of the properties of a single visual location.

(iv) A winner-take-all (WTA) network implementing the selection process based on one major rule: conspicuity of location (minor rules of proximity or similarity preference are also suggested).

(v) Inhibition of this selected location that causes an automatic shift to the next most conspicuous location. Feature maps code conspicuity within a particular feature dimension.

The saliency map combines information from each of the feature maps into a global measure where points corresponding to one location in a feature map project to single units in the saliency map. Saliency at a given location is determined by the degree of difference between that location and its surround.

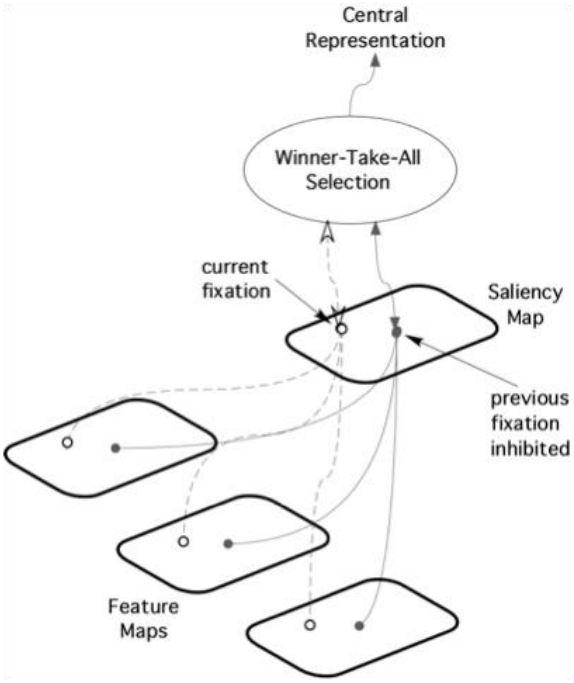


Figure 7: The Saliency Map Model as originally conceived by Koch & Ullman 1985.

2.4.2 Bottom-Up Approach

The essence of visual saliency is a bottom-up, stimulus-driven signal that announces a particular location to be sufficiently different from its surroundings so that it becomes worthy of our attention. This *bottom-up* [13] recipe of attention towards salient locations can be strongly modulated or even sometimes overridden by *top-down,* user-driven factors [11]-[12]. Thus, a lone red object in a green field will be salient and will attract attention in a bottom-up manner. The attention mechanism is led by regions of contrast in an image i.e. features like color, intensity and orientation. This does not require pre-acquired knowledge of objects and is very much task and goal independent.

2.4.3 Top-Down Approach

On the other hand, if we are looking through a bag for a red ball, amidst objects of many vivid colors, no one color may be especially salient until the top-down desire to find the red ball renders all red objects, whether balls or not, more salient.The top down approach [11]-[12] is biased on the visual processing or attention derived from the requisites of the task at hand. However, even when the target selection approach is based on top-down control, the ability of finding the target is dependent on the bottom-up stimulus factors, especially the visual similarity of targets to non-targets. The attention mechanism is led by the knowledge of the visual appearance of objects or features, etc. of the target. The context of the scene guides our attention to the regions having high chance of containing target objects. This approach is task and goal dependent [14].

References

[1] J. K. Tsotsos (1991). Is Complexity Theory appropriate for analysing biological systems?Behavioral and Brain Sciences 14(4):770-773.

[2] A. Treisman G. & Gelade (1980). A feature integration theory of attention. CognitivePsychology 12:97-136.

[3] F. Crick (1984). Function of the thalamic reticular complex: the searchlight hypothesis. Proceedings of the National Academies of Sciences USA 81(14):4586-90.

[4] E. Weichselgartner & G. Sperling (1987). Dynamics of automatic and controlled visual attention. Science 238:778-780.

[5] Schneider, Walter, and Richard M. Shiffrin. "Controlled and automatic human information processing: I. Detection, search, and attention." *Psychologicalreview* 84, no. 1 (1977): 1.

[6] Shiffrin, Richard M., and Walter Schneider. "Controlled and automatic human information processing: II. Perceptual learning, automatic attending and a general theory." *Psychological review* 84, no. 2 (1977): 127.

[7] Wolfe, Jeremy M. "Guided search 2.0 a revised model of visual search."*Psychonomic bulletin & review* 1, no. 2 (1994): 202-238.

[8] Klette, Reinhard. *Concise computer vision*. Springer, London, 2014.

[9] Shapiro, Linda G., and George C. Stockman. "Computer Vision, 2001, 279 325."

[10] Jähne, Bernd, and Horst Haußecker. *Computer vision and applications: a guide for students and practitioners*. Academic Press, 2000.

[11] Desimone, Robert, and John Duncan. "Neural mechanisms of selective visual attention." *Annual review of neuroscience* 18, no. 1 (1995): 193-222.

[12] Itti, Laurent, and Christof Koch. "Computational modelling of visual attention." *Nature reviews neuroscience* 2, no. 3 (2001): 194-203.

[13]Le Meur, Olivier, Patrick Le Callet, Dominique Barba, and Dominique Thoreau. "A coherent computational approach to model bottom-up visual attention." *IEEE Transactions on Pattern Analysis & Machine Intelligence* 5 (2006): 802-817.

[14] Awasthi, Ankit, and Keerti Choudhary. "Top Down Attentional Guidance During Visual Search."

CHAPTER 3

Video Anomaly Detection

This chapter deals with the proposed solution of the video anomaly detection problem. At first, some of the related works in this domain has been pointed out and then each component of the pipeline has been well illustrated with its theoretical background, the mathematics involved and its usage.

3.1 Related Work

Based on the definition of crowd abnormality, recent approaches [1], [7]–[9], [15], [16], [17]–[22] of anomaly detection for crowd scenes can be categorized into two classes.

1)Trajectory-based techniques:The abnormal trajectories are prone to show much lower occurring frequency than the normal ones.

2) Motion-based techniques*:*The abnormal crowd has dramatic motion patterns compared with the normal one.

For the first group of techniques, some knowledge of trajectories obtained from the normal situation is learned and then the abnormal trajectory is determined according to the learned rules [9], [17], [21], [22], [35]. For example, in Cheng and Hwang's work [9],adaptive particle sampling and Kalman filtering was used to resolve the occlusion and object segmentation error, and then the reliable trajectory types were obtained. Trajectory classification was used to localize the abnormal event.Some techniques also exploit the trajectory from particle or feature point level. For example, Wu *et al.* [1] used chaotic invariant features of Lagrangian particle trajectories to model the abnormal crowd patterns.But each representative particle here needs exhaustive tracking. In order to extract the normal/abnormal crowd patterns Cui *et al.* [8] tracked the interest points to calculate the interaction energy potentials (IEP), which explicitly exploited the relationships among a group of people. The feature representation of different patterns was analyzed and then the abnormality was declared by SVM classifier. The most direct inspiration of thispaper was proposed by Ge *et al.* [16] in the approach for crowd

structure exploitation. Each individual was tracked robustly; then the crowd groups were discovered by analyzing the relationships of trajectories. These trajectory-based methods have explicitly used high-level semantics for defining abnormality but they are always infeasible and computationally expensive for tracking each individual.

As for the second group of methods, optical flow (OF) variation [48]-[50], [7], [20]or pixel/blob change [15], [23], [52] are usually employed to explore motion patterns. Motion–pattern based methods nowadays hold the main part in the crowd anomaly detection literatures because of its ability to deal with high crowd density images. For example, a multiscale histogram of optical flow was proposed by Cong *et al.* [7] to represent the motion patterns for image sequences. The reconstruction error was computed with the trained sparse dictionary and the abnormality was detected by the motion patterns with large reconstruction cost. A streakline technique was proposed by Mehran *et al.* [49] to compute the crowd flow. The obtained streak flows were analyzed and the abnormal motion pattern was detected by a SVM predictor. Also, the social force model (SF) is another hot technique proposed recently for motion modeling in abnormal crowd detection. Through the estimation of the particle OF with SF, Latent Dirichlet allocation (LDA) [50] or other analyzers [18] can explicitly distinguish the normal/abnormal motion patterns.The mixture of probabilistic principle component analysis (MPPCA) was utilized by Kim and Grauman [37] to model the local OF. Then, the anomaly was predicted by adopting the modeled motion patterns. A spatio-temporal *Laplacian eigenmap* was proposed by Thida *et al.* [51] to extract the crowd activities. This was done by learning spatio-temporal variations of local motions in an embedded

space. An anomaly detection method was proposed by Li *et al.* [38] which were constructed by a mixture of dynamic texture (MDT) model. Spatial normalcy implemented by a center-surround discriminant saliency detector and a hierarchical model was combined for updating MDT to hierarchical MDT (H-MDT) [39]. The potential destinations and divergent centers was introduced by Wu *et al.* [20] to characterize the crowd motion in both the presence and absence of escape events. In every frame,these motion-based anomaly detection methods were usually needed to exhaustively sample image patches. The crowd context is extracted by analyzing the temporal appearance variation in these patches. Thus, this procedure also includes high computational cost.

The main emphasis of crowd anomaly detection techniques ison modeling motion patterns. But, one universal limitation of the two categories is that labeled data should be available to train the normal/abnormal pattern. However, this assumption is difficult to be satisfied in practical applications.

3.2 Overview of the pipeline

In this thesis, an anomaly crowd detection method is designed whose main components are briefly described below and the algorithm of the proposed model is given in Table 4 at the end of this chapter.

3.2.1 Pedestrian Detection:

If a video sequence is given, the thesis uses the pedestrian detection algorithm proposed by Dalal *et. al.*[10], which uses Histogram of Gradients

(HOG) and Support vector machine (SVM) for extracting each target of the crowd and thenthe obtained targets are marked by rectangular regions of different sizes.

3.2.2 Structural Context Description:

The visual contextual information of the individuals in the crowd are extracted by a structural contextual descriptor (SCD) described in the work of Yuan *et. al.* [33]. The PEF-PIF model in solid-state physics is used to establish the relation between an examined target and its neighbors. This relationship helps in the determination of a weight measuring the motion difference computed by SHOF.

3.2.3 Saliency Motion Description:

The motion context of the individuals is extracted from the saliency values of the temporal saliency maps obtained from a superpixel level saliency model using local graph construction and graph matching. The weight between an examined target and its neighbors is then calculated by Hausdorff distance metric.

3.2.4 Object Association using 3D-DCT:

It is difficult to explicitly model the appearance change for the target association. The work is however not of tracking, but of associating targets and finding stable individuals called *observers*. This context based object tracking shows the way for the SCD and SMD variation computation.

3.2.5 Fusion of SCD and SMD:

The motions context or weights obtained by each of these two methods are fused using a fusion technique thus providing information of motion of an individual using two cues namely optical flow and saliency map. The fused weight and the fused feature vector are then used for computing the SCD and SMD variation of the observers obtained from 3D-DCT model.

3.2.6 Anomaly Detection:

This step is used for detecting anomaly by the spatial and temporal analysis of SCD and SMD variation. The number of targets in each frame may not be same always; hence it causes different SCD dimensions in adjacent frames. Thus, Earth mover's distance (EMD) [28] is used to compute the SCD and SMD variations because it can analyze the similarity of two distributions with different dimensions.

3.3 Pedestrian Detection

Pedestrian detection is an essential and important task in any intelligent video surveillance system, as it provides the basic information for semantic understanding of the video footages. It has an extensive application in automotive applications due to its potential for improving safety systems. Pedestrian detection is a key problem in computer vision, and has several applications including robotics, surveillance and automotive safety [2].

Detecting people in images is a problem that has been a significant topic of research for a long time [3]-[6], [10]. In the recent past, there has been a surge of interest in pedestrian detection [11]-[14]. Pedestrian detection techniques that can produce more accurate results would have immediate and far reaching impact to applications such as surveillance, robotics, assistive technology for the visually impaired, content based indexing, advanced human machine interfaces and automotive safety.

There are various challenges that hinder the pedestrian detection techniques such as various style of clothing in appearance, different possible articulations, presence of occluding objects, and frequent occlusion between pedestrians.Despite the challenges, pedestrian detection is still an active research area in computer vision in recent years. Numerous approaches have been proposed over the years for pedestrian detection.

In this thesis the model used for pedestrian detection is training the detector using local feature descriptor viz. histogram of oriented gradients (HOG) and then a classifier viz. support vector machines (SVM) is used to classify the objects according to the features obtained.

3.3.1 Histogram of Gradients

Histogram of oriented gradients (HOG) is a feature descriptor used in computer vision and image processing fields for the purpose of object detection. The occurrences of gradient orientation in localized portions of an image are counted by this technique. The main concept behind the histogram of oriented gradients descriptor is that the local object appearance and shape within an image can be depicted by the distribution of intensity

gradients or edge directions. The image is first divided into small regions that are connected and these are called cells. A histogram of gradient directions is compiled for the pixels within each cell. The final descriptor is the concatenation of these histograms. The contrast of the local histograms can be normalized by calculating a measure of the intensity across a larger region of the image, called a block. This normalized contrast value is then used to normalize all cells within the block, thus improving accuracy. This normalization provides better invariance to changes in illumination and shadowing.

The HOG descriptor holds key advantages over other descriptors. As it operates on local cells, it is invariant to photometric and geometric transformations, except for object orientation. These changes would only be accountable in large spatial regions. As long as the pedestrians maintain a roughly upright position the coarse spatial sampling, fine orientation sampling, and strong local photometric normalization allows the individual body movement of pedestrians to be ignored. The HOG descriptor is thus very much suited particularly for human detection in images [10].

The different steps involved in extracting features using HOG are:

(a) Gradient Computation: In any image pre-processing, the first step of computation in many feature detectors is to ensure normalized color and gamma values. However, this normalization step can is not required in HOG descriptor computation, as Dalal and Triggs pointed out in their work [10] that the ensuing descriptor normalization essentially achieves the same result. Thus the performance of HOG descriptor is hardly affected by image pre-processing. The first step of HOG descriptor calculation is the

computation of the gradient values. The basic method is to apply the 1-D centered, point discrete derivative mask in the horizontal direction or vertical direction or both. This method basically requires filtering the color or intensity data of the image with the following filter kernels: $\begin{bmatrix}-1 & 0 & 1\end{bmatrix}$ and $\begin{bmatrix}-1 & 0 & 1\end{bmatrix}^T$.

Several other complex masks such as 3x3 Sobel mask or diagonal masks can also be used but these masks performs poorly in detecting humans in images [10]. Gaussian smoothing can also be applied ahead of the derivative mask, but it was found that it performed better without any smoothing term [10].

(b) Orientation binning: The next step is creation of the cell histograms. Based on the values found in the gradient computation each pixel within the cell gives a weighted vote for an orientation-based histogram channel. The cells can either be rectangular or radial in shape. The histogram channels are evenly spread over 0 to 180 degrees if the gradient is "unsigned" or over 0 to 360 degrees if the gradient is "signed". The best performance is achieved in human detection experiments if unsigned gradients are used with 9 histogram channels [10]. As far as the vote weight is considered, each pixel contributes either the gradient magnitude itself, or some function of the magnitude. However, best results are obtained when the gradient magnitude itself is used. Many other options are available for casting vote that includes the square root or square of the gradient magnitude, or even some clipped modification of the magnitude [10].

(c) Descriptor Blocks: Each cell consists of a group of pixels; these cells are then grouped together into larger, spatially connected blocks so that the gradient strengths are

locally normalized to account for the changes in illumination and contrast. In this way the components of the normalized cell histograms are obtained from all the block regions. The concatenated vector of all these components forms the HOG descriptor. Typically, these blocks overlap thus ensuring that each cell contributes more than once to the formation of the final descriptor. There exist two main block geometries: rectangular R-HOG blocks and circular C-HOG blocks.In this thesis rectangular R-HOG blocks have been used. R-HOG blocks are basically square grids, presented by three parameters: the number of pixels per cell, the number of pixels per cell, and the number of channels per cell histogram. For human detection the optimal parameters are found to be four 8x8 pixels cells per block, 4 cells constitute a block i.e. 16x16 pixels per block and 9 histogram channels [10]. In order to weight the pixels less around the edge of the blocks, a Gaussian spatial window is applied within each block before calculating the histogram votes which gives a little improvement in the performance.

(d) Block Normalization: The final step is the block normalization. There are four different methods used for block normalization [10]. Let the non-normalized vector containing all histograms in a given block be denoted by v, the k-norm of v be denoted by $\|v\|_k$ for $k = 1, 2$ and e is some small constant. The normalization factor f can then be defined by any one of the following:

L2-norm: $f = \frac{v}{\sqrt{\|v\|_2^2 + e^2}}$

L2-hys: Clipping the L2-norm such that maximum values of v be limited to 0.2 and normalizing again [36].

L1-norm: $f = \frac{v}{\left(\|v\|_1 + e\right)}$

L1-sqrt: $f = \sqrt{\frac{v}{\left(\|v\|_1 + e\right)}}$

Experimentally, it is found that the L2-hys, L2-norm, and L1-sqrt schemes give similar performance, while the L1-norm gives slightly less reliable performance. Generally, all of the four methods give significant improvement over the non-normalized data [10].

Thus using HOG feature descriptor, the features are extracted from the images. Now, we need to classify the features of an object to be detected (positive samples) and that of the background or objects of no interest (negative samples). To complete this classification step, a well-known classifier viz. Support Vector Machine (SVM) is employed which is explained in the next section.

3.3.2 Support Vector Machine

Support Vector Machines (SVM) is a maximum margin classifier which tries to maximize the distance between the hyper plane and nearest training data points or also known as support vectors.

Let us consider two classes which are levelled as +1/-1. The samples denoted by $Z = \left\{x^t, r^t\right\}$ where, x^t are the observed data points and r^t are the corresponding class

level (+1/-1). The objective or training phase of the SVM classifier is to find an optimal hyper plane which best separates the training vector and also maximizes the margin between hyper plane and nearest training data points (support vectors) on the either side of the plane for better generalization.

Let the hyper plane is denoted by,

$$\vec{w}^T \vec{x} + w_o \tag{1}$$

In training phase it is necessary to find the weight vector $\vec{w}$ and the bias w_o such that

$$\begin{aligned} \vec{w}^T \vec{x} + w_o \geq +1, \quad & if \quad r = +1 \\ \vec{w}^T \vec{x} + w_o \leq -1, \quad & if \quad r = -1 \end{aligned} \tag{2}$$

.

The optimization problem can be constructed as,

$$\text{Minimize} \qquad J(\vec{w}, w_o) = \frac{1}{2}\vec{w}^T \vec{w} + \frac{1}{2} w_o^2 \tag{3}$$

$$\text{subject to the constraints} \qquad r^t \left(\vec{w}^T \vec{x}^t + w_o \right) \geq +1, \quad \forall t \tag{4}$$

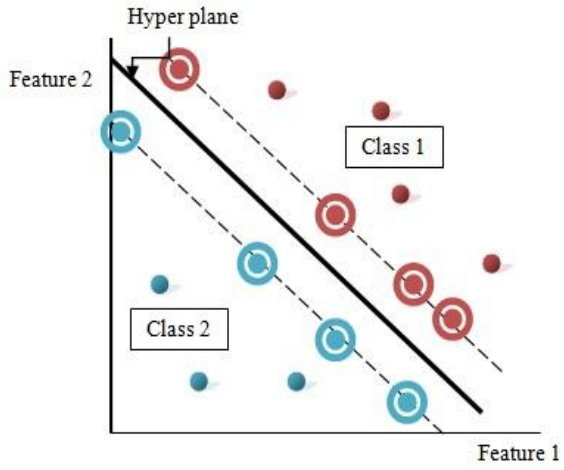


Figure 8: A Linear SVM Classifier

Figure 8 shows the classification strategy of SVM classifier with linear hyper plane for features vectors with two feature entities for two-class problem. The data points on the dotted lines are called support vectors. The distance between two dotted lines is the margin. For multi-class problem, "*one verses all*" strategy is used and the number of hyper plane obtained is same as the number of the class present in the training dataset.

The desired classifier is realized using liblinear support vector machine [24], a publiclyavailable Matlab version of SVM. We adopted linear kernels as they are faster and performs comparably well as non-linear polynomial and also the computational complexity of the training is reduced by using the linear kernels.

Hence the problem of pedestrian detection is solved by using HOG and SVM, and this method provided reasonable results for moving further in solving the main problem of anomaly detection. The pedestrian detection results are discussed in the Results section of the next chapter.

3.4 Motion Context using Optical Flow

After the pedestrians have been detected, the next step is to build a crowd structure representation. The crowd consists of many non-isolated individuals who have some connection with each other [25]-[26]. Thus the crowd structure can be formed from these relations.Yuan *et. al.* proposed a new structural contextual descriptor (SCD) [33] to represent the structure of each individual. The notion behind this SCD representation was that if individuals show large behavior difference with their neighbors then they are highly probable to be abnormal. An inconsistency weight is used to represent the variance between the examined target and its surroundings. A large weight represents that the target's behavior is more distinguishable to its surroundings. Yuan *et. al.* introduced the concept of potential energy function of particle's interforce (PEF-PIF) in solid-state physics [33], then the generation of selective histogram of optical flow (SHOF) is implemented and finally the SCD computation is done in connection with the PEF-PIF model.

3.4.1 Potential Energy Function of Particle's InterForce:

The fundamental description of the potential energy of two particles can be expressed as:

$$U(r) = \frac{a}{r^m} - \frac{b}{r^n} \tag{5}$$

where, $U(r)$ is the potential energy of the two particles, r is their Euclidean distance, and *a, b, m and n* (generally m>n) are the empirical constants. The first term and the second term in the right hand side represents rejecting potential energy field and the attracting potential energy field respectively. When *r* is small the two particles exhibit rejecting state and when *r* is large they exhibit attracting state.

The force of two particles can be defined by combining power (CP) which is the negative deviation of $U(r)$ and is given as

$$f(r) = -\frac{dU(r)}{dr} = \frac{ma}{r^{m+1}} - \frac{nb}{r^{n+1}} \tag{6}$$

Also the linking weight of the two particles $w(r)$ with distance *r* is given as

$$w(r) = \frac{1}{|f(r)|} \Big/ \left(\int_{\sqrt{2}}^{r} \frac{1}{|f(r)|} dr \right), \qquad r \in \left[\sqrt{2}, \infty \right] \tag{7}$$

3.4.2 Selective Histogram of Optical Flow (SHOF) generation:

The basic information contained in a video can be extracted through motion context, and for that optical flow (OF) [47] is utilized to qualify the motion of every individual. The pedestrian detection algorithm gives a bounding box marking each individual. Hence, histogram of optical flow (HOF) [46] is used to calculate the motion patterns of each individual where each bin represents the direction of OF and the value of each bin is the magnitude of the OF corresponding to that direction. Within a single individual, the pixel's motion directions are highly uniform which helps to build the idea

that the magnitude property of an individual can be represented by the maximum of HOF of that individual. The notion of anomaly is different for different crowded scenes, such as magnitude inconsistency where magnitudes are important or direction inconsistency where directions are important.

From a few initial frames of the crowded scenes, a parameter ξ is learned which is used to limit the range of HOF. This limited HOF is called SHOF which is required for motion difference computation. Let $\mathbf{H}_0^{\xi}$ be the SHOF of the examined target and $\mathbf{H}_i^{\xi}$ be that of the i^{th} neighbor. χ^2 distance is used to compute the difference Δf between them and is given by

$$\Delta f_i = \chi^2\left(\mathbf{H}_0^{\xi}, \mathbf{H}_i^{\xi}\right) \tag{8}$$

where,
$$\chi^2\left(h_1, h_2\right) = \frac{1}{2}\sum_{i=1}^{B}\frac{\left|h_{1;i} - h_{2;i}\right|^2}{h_{1;i} + h_{2;i}} \tag{9}$$

and B is the number of histogram bins.

Let M be the number of neighbors around the examined target, then a vector $\Delta\mathbf{f}$ of all the motion differences (Δf_i's) are obtained for each target where *i* ranges from 1 to M, and it is denoted as $\Delta\mathbf{f} = \{\Delta f_1, \Delta f_2, \Delta f_3, ..., \Delta f_M\}$. Such a vector $\Delta\mathbf{f}$ is obtained for all the individuals in a frame.

A small ξ means a narrow range of SHOF, while a larger ξ means wider range of SHOF. If the magnitude inconsistency is of more importance then ξ is set as small as

possible; whereas if the motion direction of individual is of concern then ξ is set large so that more bins are considered which represents the direction of OF. The optimal ξ is learned from the motion difference of the individuals in the normal video frames, and given as

$$\hat{\xi} = \arg\min_{\xi} Var\left(\Delta \mathbf{f}\right) \tag{10}$$

Where *Var(.)* means calculation of variance. The equation (10) is solved by varying ξ in the range [0, 1] with an interval of 0.1. The initial frames of a video are generally considered normal and hence are used for learning $\hat{\xi}$. As $\hat{\xi}$ is obtained it is used as a parameter for subsequent video frames to select the range of histogram of optical flow.

3.4.3 SCD calculation

The contribution of all the neighbors should be taken into account while calculating the contextual structure of an examined target. Hence, the detailed equation (7) is normalized based on the number of neighbors or surrounding individuals and is rewritten as

$$w\left(\Delta f_i\right) = \frac{1}{f\left(Z\Delta f_i\right)} \Big/ \left(\sum_{i=1}^{M} \frac{1}{f\left(Z\Delta f_i\right)}\right) \tag{11}$$

where Z is a constant used to enlarge Δf to a reasonable range so that equation (7) can be used. Δf can be seen as an analogy to r in equation (7) and $w\left(\Delta f_i\right)$ is the weight

between the examined target and the i^{th} neighbor and M is the number of neighbors around the target.

Let $W_{OF} = \{\mathbf{W}_k\}_{k=1}^{M+1}, \mathbf{W}_k \in \mathbb{R}^{1\times M}$ is the weight vector of the k^{th} individual which can be calculated by (11) and $F_{OF} = \{\mathbf{F}_k\}_{k=1}^{M+1}, \mathbf{F}_k \in \mathbb{R}^{4\times M}$ is the corresponding feature vector of the *M* neighbors, whose column elements are max, min, mean and variance of motion energy in the individual bounding box. The motion difference of all the individuals with their neighbors are calculated and the SCD is established which is denoted as $\{W_{OF}, F_{OF}\}$.

3.5 Motion Context using Saliency

In this section the motion context of a video is extracted using saliency, specifically saliency maps generated from superpixel level calculation. Basically, the work is to detect the superpixels that are in motion. The proposed method of saliency is a bottom-up approach. Each individual video frame is first over segmented to generate a group of pixels, called superpixels and finally these superpixels are fused together to produce the final segmentation. Simple Linear Iterative Clustering (SLIC) [38] is applied for superpixel generation in each video frame sequentially when a particular frame is processed. Some representative superpixels are chosen as superseeds for temporal graph matching. A superseed is extracted from each segment in the finally segmented previous video frame. Labels of these superseeds are propagated to the current frame from the previous frame by using local graph matching.

3.5.1 Initialisation

The extraction of superpixels in each frame of the video is done by the well-known SLIC [38] algorithm. The superpixel segmentation of the frames is given by:

$$S_t = SLIC(t,k) \tag{12}$$

where, St denotes the superpixels extracted from the current frame t and k is the desired number of superpixels to be generated. Superpixel level direct frame difference (SDFD) is used to find the superpixels which can potentially be in motion. The application of SDFD needs the values of the difference in the mean intensity values of the co-located superpixels in the current and previous frames.

Mathematically,

$$SDFD_{S_{i,t}} = g\left(S_{i,t}\right) - g\left(S_{i^*,t-1}\right) \tag{13}$$

where, $g\left(S_{i,t}\right)$ is the the mean intensity of $S_{i,t}$, the i^{th} superpixel in the current frame *t* and $g\left(S_{i^*,t-1}\right)$ is the mean intensity of $S_{i^*,t-1}$, the co-located superpixel (i^*) in the previous frame (*t-1*).

The term $SDFD_{S_{i,t}}$ represents the difference in the two intensity values for the superpixel $S_{i,t}$ in the current and previous frame, and is compared with an experimentally chosen threshold (T_1). The superpixels, for which $SDFD_{S_{i,t}}$ are larger

than T_1 are considered to be in motion. The superpixels found to be in motion are marked by motion labels $\varnothing_{S_{i,t}}$ with value 1. Hence, a binary motion map is built given by:

$$\varnothing_{S_{i,t}} = \begin{cases} 1 & if \quad SDFD_{S_{i,t}} \geq T_1 \\ 0 & else \end{cases} \tag{14}$$

So, the binary motion map for a frame f_t is the union of the binary motion maps of all the superpixels of that frame and can be expressed as

$$\varnothing_t = \bigcup_{i=1}^{n} \varnothing_{S_{i,t}} \tag{15}$$

where n denotes the total number of superpixels in the frame f_t.

3.5.2 The Saliency Model

Saliency detection deals in extracting regions which acquire greater attention of human vision [39]. There are many pixel based saliency models in spatial domain [40] [41] and also in frequency domains [39] [43]. Liu et. al. [42] in his work proposed a superpixel level spatio-temporal saliency detectionalgorithm which uses pixel level optic flow for superpixel motion estimation. This method is efficient in terms of accuracy but suffers from low computational efficiency. In this thesis motion information is extracted fromthe temporal saliency map of the proposed superpixel based saliency model which is computationally efficient.

Temporal saliency detection:

In the proposed saliency model the superpixels are generated for each frame and temporal saliency for only those superpixels are considered which have non-zero motion in the current frame. Local region graphs are used for determining temporal matching between these superpixels in the current frame and the co-located superpixels in the previous frame. The detailed steps are explained below:

(a) ***Local graph construction:***

$G_{S_{i,t}}(v,e)$ and $G_{S_{i^*,t-1}}(v,e)$ are the local region graphs built surrounding the superpixels i in motion in the current frame t and the collocated superpixels i^* in the previous frame (*t-1*) respectively. The vertices of each of the graphs $G_{S_{i,t}}(v,e)$ and $G_{S_{i^*,t-1}}(v,e)$ are the neighboring superpixels of $S_{i,t}$ and $S_{i^*,t-1}$ respectively. In each of these graphs, edges are defined between the center vertex and each of the neighbors, and also between each pair of neighbor vertices. The graphs are shown in figure below. The spatial affinity between any two superpixels is calculated which is the product of color (only Luminance value *L* for gray images) and texture affinities between them. This value obtained is set as the corresponding edge weight of those superpixels. The color affinity $C(S_m; S_n)$ and texture affinity $T(S_m; S_n)$ between superpixels $S_m$and S_n are given by

$$C(S_m, S_n) = \left\| \overline{Lab_{S_m}} - \overline{Lab_{S_n}} \right\|^2 \tag{16}$$

$$T\left(S_m, S_n\right) = W_H\left(SLBP_{P,R}\left(S_m\right) \oplus SLBP_{P,R}\left(S_n\right)\right) \tag{17}$$

Hence, the edge weight between the two superpixels S_m and S_n is given by:

$$w_{S_m,S_n} = C\left(S_m, S_n\right) * T\left(S_m, S_n\right) \tag{18}$$

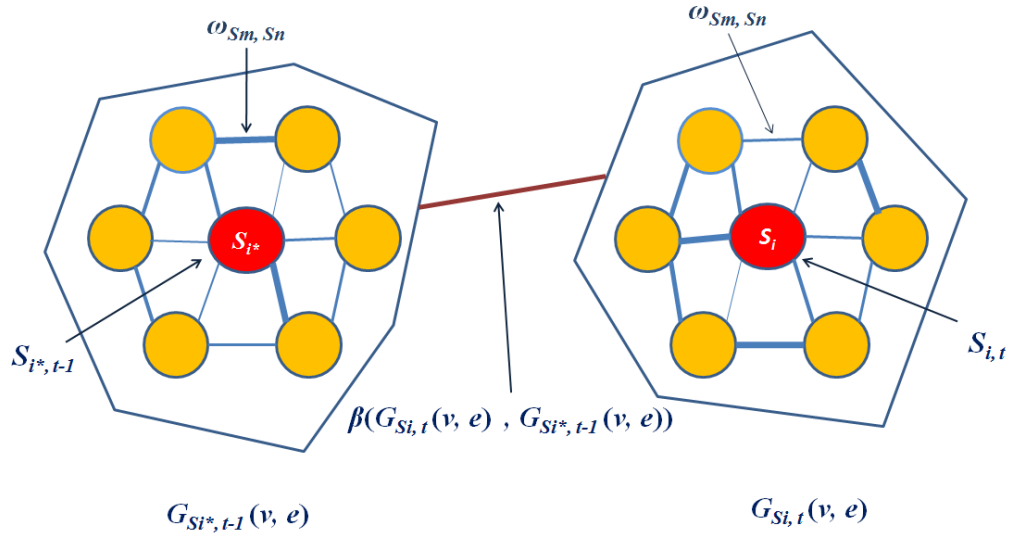


Figure 9:Local Graph Matching

(b) Local graph matching:

There are many graph matching techniques as proposed in the works of [44]. The work of [45] is employed for the purpose of graph matching. Let A_1, A_2 be the adjacency matrices; D_1, D_2 be the diagonal matrices and L_1, L_2 be the Laplacian matrices of the

graphs $G_{S_{i,t}}(v,e)$ and $G_{S_{i^*,t-1}}(v,e)$ respectively. Then, the Laplacian matrices are given by:

$$L_1 = D_1 - A_1 \tag{19}$$

$$L_2 = D_2 - A_2 \tag{20}$$

The dissimilarity score β between the two graphs is given by the differences of top K eigenvalues $(\lambda_{11}.....\lambda_{1K})$ of L_1 and $(\lambda_{21}.....\lambda_{2K})$ of L_2. So, we can write:

$$\beta = \sum_{k=1}^{K} (\lambda_{1k} - \lambda_{2k})^2 \tag{21}$$

90% of the energy is contained in the top K eigenvalues. Thus, K is determined using the following equation as:

$$\min_{q\in[1,2],p} \left(\frac{\sum_{p=1}^{K} \lambda_{qp}}{\sum_{p=1}^{M} \lambda_{qp}} > 0.9 \right) \tag{22}$$

where, M is the total number of eigenvalues.

The higher is the value of β, the higher is the dissimilarity between the graphs.

(c) The temporal saliency map:

An experimentally chosen threshold is chosen to check the temporal saliency and denoted as T_2. The dissimilarity value β between the two graphs $G_{S_{i,t}}(v,e)$ and $G_{S_{i^*,t-1}}(v,e)$ centering two co-located superpixels S_i and S_{i^*} is compared with T_2. If the

dissimilarity value β is less than T_2, then the temporal saliency value of the superpixel (S_i) in the current frame *t*is the temporal saliency value of the co-located superpixel (S_{i^*}) from the previous frame (*t-1*). Otherwise, the dissimilarity β value itself is set as the saliency value of the superpixel under consideration. Finally, the temporal saliency map for the frame $f_t$can be expressed as:

$$\Omega_t = \bigcup_{i=1}^{n} \Omega_{S_{i,t}} \tag{23}$$

3.5.3 The Saliency Motion Descriptor (SMD) computation:

As we have generated the temporal saliency map for each frame, the saliency values of each pedestrian in a frame are extracted from those maps. These saliency values signify the motion values of each pedestrian in a frame. As the number of pixels defining each individual in a frame is different from one another, Hausdorff distance metric [37] is used to generate the weight between each individual and its corresponding neighbor. The Hausdorff distance between two matrices *X* and *Y* is mathematically defined as:

$$d_H(X,Y) = \max\left\{\left(\max_{x\in X} \min_{y\in Y} d(x,y)\right), \left(\max_{y\in Y} \min_{x\in X} d(y,x)\right)\right\} \tag{24}$$

where, $d(x,y)$ is the Euclidean distance between pixel $x \in X$ and pixel $y \in Y$.

Let $W_{SAL} = \{\mathbf{W}_k\}_{k=1}^{M+1}, \mathbf{W}_k \in \mathbb{R}^{1\times M}$ is the weight vector of the k^{th} individual given by $d_H(X,Y)$ and $F_{SAL} = \{\mathbf{F}_k\}_{k=1}^{M+1}, \mathbf{F}_k \in \mathbb{R}^{4\times M}$ is the corresponding feature vector of the

*M*neighbors, whose column elements are max, min, mean and variance of the motion energy in the individual bounding box of the saliency map. The motion difference of all the individuals with their neighbors are calculated and the SMD is established which is denoted as $\{W_{SAL}, F_{SAL}\}$.

3.6 3-D DCT

A visual tracking system generally needs an object appearance model that is robust to changing illumination, pose and other factors that may be encountered in a video. There are many multi-object trackers available [31]-[32] but they are difficult to be implemented in the crowd anomaly detection because of the high density of the crowd, illumination changes and frequent occlusion. Besides, these trackers are computationally expensive. Many trackers also take help of the appearance samples in previous frames to form the basis on which the object appearance model has been built. This approach suffers limitations that the bases are driven by the input data, which can be easily corrupted and also the updating of the bases is difficult in challenging situations. Thus, an appearance model is built using the 3D-discrete cosine transform (3D-DCT) because the 3D-DCT is based on a set of cosine basis functions that are determined by the dimensions of the 3D signal and are thus independent of the input video data [30]. Also, the 3D-DCT generates a compact energy spectrum which has sparse high-frequency components if the appearance samples are similar. So, by discarding these high-frequency components, a compact 3D-DCT based object representation is obtained. To update the object representation efficiently, an incremental 3D-DCT algorithm is used which decomposes

the 3D-DCT into continuous operations of the 2D discrete cosine transform (2D-DCT) and 1D discrete transform (1D-DCT) on the input video data [30]. Thus the 3D-DCT algorithm only requires computing the 2D-DCT for the newly added frames and the 1D-DCT along the third dimension. This significantly reduces the computational complexity. After this incremental 3D-DCT algorithm is established, a discriminative criterion is designed to evaluate the likelihood of a test sample belonging to the foreground object.

However, the research work done in this thesis is not of tracking every target but to identify the abnormality. This task can be completed by considering only the stable targets called *observers* [33]. For these individuals, occlusion and appearance/illumination change hardly occur and they can be well tracked. Thus by analyzing the observers' temporal SCD variation and the temporal SMD variation the abnormality can be robustly detected. Thus this method is different from the conventional multi-object trackers. Here, the 3-D DCT is used only to seek the stable observers [33]. A newly proposed 3-D DCT model [30] is employed which has an excellent ability of incremental analysis. The designed 3-D DCT multi-object tracker consists of three components: compact 3- D DCT template representation, multi-target association, and incremental template updating. Each component is described sequentially as follows.

3.6.1 Compact 3-D DCT-Based Object Representation

In a video sequence, to formulate the target association the frames are considered as a 3-D volume by concatenating them. Then the self-correlation of the recently observed target sample is incrementally evaluated with the previously collected target sample set. The 3-D DCT [30] is utilized as a tool to complete this task.

In a given video sequence, the previously collected sample set can be assumed to be denoted as $\left(s_{\mathbf{III}}(x,y,z)\right)_{N_1\times N_2\times N_3}$, where $N_1$and N_2 are the width and height of the sample, and N_3 is the number of samples in the target sample set. The new target sample in the next frame is denoted as $\left(n(x,y)\right)_{N_1\times N_2}$. $\left(s'_{\mathbf{III}}(x,y,z)\right)_{N_1\times N_2\times(N_3+1)}$ is denoted as the concatenated target sample set, where the $(N_3+1)^{th}$frame is chained to the end of the previous target sample set. According to the 3-D DCT [30], $\left(s'_{\mathbf{III}}(x,y,z)\right)_{N_1\times N_2\times(N_3+1)}$ can be represented as

$$\begin{aligned} S' &= \mathbf{C}_{\mathbf{III}} \times_1 \mathbf{D}_1^{\,T} \times_2 \mathbf{D}_2^{\,T} \times_3 \left(\mathbf{D}'\right)_3^{\,T}, \\ \mathbf{C}_{\mathbf{III}} &= S' \times_1 \mathbf{D}_1 \times_2 \mathbf{D}_2 \times_3 \left(\mathbf{D}'\right)_3, \end{aligned} \tag{25}$$

where $S' = \left(s'_{\mathbf{III}}(x,y,z)\right)_{N_1\times N_2\times(N_3+1)}$, $\mathbf{C}_{\mathbf{III}} \in \mathbf{R}^{N_1\times N_2\times(N_3+1)}$ represents the 3-D DCT coefficient matrix, and $\times_m$ is the mode-*m* product defined in tensor algebra [34].

$\mathbf{D}_1 = \left(a_1(o,x)\right)_{N_1\times N_1}$ is a cosine basis matrix whose elements are represented as

$$a_1(o,x) = a_1(o)\cos\left(\frac{\pi(2x+1)o}{2N_1}\right) \tag{26}$$

$\mathbf{D}_2 = \left(a_2(p,y)\right)_{N_2\times N_2}$ is a similar cosine basis matrix whose elements are specified as

$$a_2(p,y) = a_2(p)\cos\left(\frac{\pi(2y+1)p}{2N_2}\right) \tag{27}$$

and $\left(\mathbf{D}'\right)_3 = \left(a_3'\left(q,z\right)\right)_{(N_3+1)\times(N_3+1)}$ is a different cosine basis matrix whose elements are denoted as

$$a_3'\left(q,z\right) = \begin{cases} \sqrt{\dfrac{1}{N_3+1}}, & if \quad q=0 \\ \sqrt{\dfrac{2}{N_3+1}}\cos\left(\dfrac{\pi\left(2z+1\right)q}{2\left(N_3+1\right)}\right), & otherwise \end{cases} \tag{28}$$

where, $o \in \{0,1,...,N_1-1\}$, $p \in \{0,1,...,N_2-1\}$, $q \in \{0,1,...,N_3-1\}$, and $a_k\left(o/p/q, x/y/z\right)$ is defined as

$$a_k\left(o/p/q, x/y/z\right) = \begin{cases} \sqrt{\dfrac{1}{N_k}}, & if\ o/p/q=0 \\ \sqrt{\dfrac{2}{N_k}}, & otherwise \end{cases} \tag{29}$$

The properties of 3-D DCT indicate that the larger the values *(o, p, q)* are,the higher is the frequency which the corresponding component of $\mathbf{C_{III}}$ encodes. Based on the values of*(o, p, q)*, the work [30] shows that the high frequency coefficients are usually sparse (e.g., texture clue) and are hence removed. The low-frequency coefficients are relatively dense (e.g., mean value) and are preserved. The component $\mathbf{C_{III}}$ gets compressed as a result. Therefore, the compact 3-D DCT object representation $\mathbf{C_{III}}$ is modified as:

$$\mathbf{C}_{\mathbf{III}}^{*} = S^{*} \times_1 \mathbf{D}_1 \times_2 \mathbf{D}_2 \times_3 \left(\mathbf{D}'\right)_3, \tag{30}$$

where $S^* = \left(s^*_{\mathbf{III}}(x,y,z)\right)_{N_1 \times N_2 \times (N_3+1)}$ is the approximation of S' representing its corresponding reconstructed image sequence. Based on this compression, a reconstruction error representing the loss of low-frequency components is involved, which is defined as:

$$e = \left\| n - s^*_{\mathbf{III}}(:,:,N_3+1) \right\|^2 \tag{31}$$

For a new target sample, the consistency likelihood is measured with the target sample set which is given as:

$$L = \exp\left(-\frac{1}{2\lambda} e\right) \tag{32}$$

where λ is the scaling factor, and its typical value is 0.1 [33].

This likelihood measurement is used for target association in Section 3.6.2 and also for the modeling of incremental template updating which is described in Section 3.6.3.

3.6.2 3-D DCT-Based Multi-Target Association

The 3-DDCT is utilized to associate the targets in different frames because of its effectiveness in representing a video sequence. Now, we need to match each newly detected pedestrian with a previously constructed template pool or target set. To strengthen the accuracy of the target association, we should consider not only the appearance consistency, but also the target neighborhood. Figure 10illustrates the

flowchart which has two constraints, namely, the appearance consistencyand the neighborhood smoothness. Some mathematical notations and their meanings are presented in Table 1 which would be required in subsequent discussion.

Table 1: Some mathematical symbols in target association

Symbols	Meaning
t	Time index
M	Number of target samples in each frame.
$\left\{T_T^f\right\}_{f=1}^F$	F template pools of target appearance
$\left\{T_C^f\right\}_{f=1}^F$	F template pools of target neighborhood
$\left\{n_{t+1}^i\right\}_{i=1}^M$	M new samples at time t+1
$\left\{n_{C_{t+1}}^i\right\}_{i=1}^M$	M new neighborhood samples at time t+1

1) Appearance Consistency: Appearance consistency works under the assumption that the appearance of the new target should match its corresponding template pool of appearance. Each target appearance n_{t+1}^i is compared with each template pool of appearance T_T^f . The reconstruction error $e_{t+1}^{i,f}$ is thuscomputed by equation (31). The consistency of the i^{th} target appearance with f^{th} template pool is denoted as:

$$L_{T_{t+1}}^{i,f} = \exp\left(-\frac{1}{2\lambda} e_{t+1}^{i,f}\right) \tag{33}$$

The appearance consistency is directly related to $L_{T_{t+1}}^{i,f}$, i.e., the larger the value of $L_{T_{t+1}}^{i,f}$ is, the more consistent the appearance is with the template pool.

2) Neighborhood Smoothness: The neighborhood smoothness is also a constraint for target association. A surrounding rectangular region is drawn around the target considering it as the center. This rectangular region represents the neighborhood of the target. Let $L_{C_{t+1}}^{i,f}$ denotes the smoothness of the i^{th} target neighborhood with that of the f^{th} template pool. The strategy of inference is the same as the appearance consistency.

Considering the above two constraints, the target sample that is most closely related to the f^{th} template pool is defined as

$$\bar{n}_{t+1}^{f} = \arg \max_{n_{t+1}^{i}} \; Normalize\left(L_{T_{t+1}}^{i} \cdot L_{C_{t+1}}^{i}\right) \tag{34}$$

where,*Normalize(·)* is the function normalizing the $\left\{L_{T_{t+1}}^{i} \cdot L_{C_{t+1}}^{i}\right\}_{i=1}^{M}$ into [0, 1] and *M* is the number of neighbors. If *Normalize(·)* >0.8, then the examined target is declared as an observer; or else, it is discarded and fails in the target association. Simultaneously, the corresponding template pool is updated by:

$$\begin{aligned} T_T^f &= Concatenate\left(T_T^f, \bar{n}_{t+1}^f\right) \\ T_C^f &= Concatenate\left(T_C^f, \bar{n}_{C_{t+1}}^f\right) \end{aligned} \tag{35}$$

where,*Concatenate(·)* is the function concatenating the newly obtained target sample with its related template pool.

After the association is completed, each target sample along with its contextual sample is added to its corresponding template pool and simultaneously the template pool is updated.However the size of the template pool cannot increase outside a limit due to increasing computational complexity.Thus a maximum threshold is set for association such that the redundant sample is discarded, if necessary.

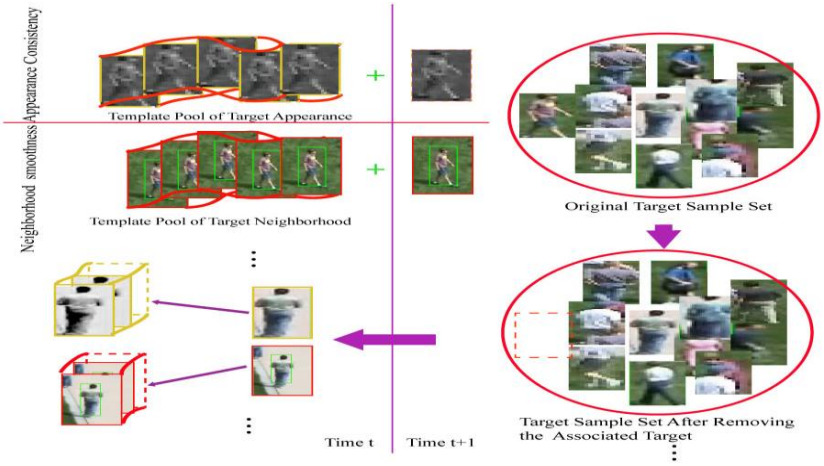


Figure 10: Representation of multi-target association.

3.6.3 Incremental Template Updating

There are two factors that need to be optimized for thetemplate updating including template pools of target appearance and the neighborhood.

1) The first factor is the reliability with the previously constructed template pool and it isdesired that the updated template contains more information than in the previous frames.

2) The second factor is the adaptability for the dynamic scene. Unlike reliability factor, it is desired that the updated template changes adaptively with the dynamic scene.

Generally speaking, in a video sequence, the targets in the initialframes do not undergo substantial changes in appearance or with the neighborhood. In contrast,the targets in the later frames are prone to change. Thus a two-stage updating strategy is used [33], where, reliability is considered as the main criterion in the beginningstages and adaptability is consideredas the principle criterion in the later stages of the tracking process.

1) Reliability Preservation:Let *K* be the fixed number which defines the maximum capacity of samples of each template pool. If the size of the new f^{th} template pool, T_f is smaller than *K*, then T_f is kept unchanged and if its size exceeds K, i.e., reaches *K*+1, then the most dissimilar target is removed from the template pool. To maintain this condition, thesimilarity between each target sample n_k^f, $k \in [1, K]$ within T_f and the remaining $s_{\mathbf{III}}^f(:,:,1:K)$ (which is actually $\{T_f - n_k^f\}$) is iteratively evaluated according to the reconstruction error in Section 3.6.2. The removed target sample $\bar{n}^f$ is selected by

$$\bar{n}^f = \arg \min_k \exp\left(-\frac{1}{2\lambda}\left\| n_k^f - \left(s_{\mathbf{III}}^f\right)^*(:,:,K)\right\|^2\right) \quad (36)$$

where, $\left(s_{\text{III}}^{f}\right)^{*}(:,:,1:K)$ is the obtained 3-DDCT model.

2) Adaptability Preservation: In case of the adaptability preservation, it is desirable that the template pool gives more priority to the newly observed sample than the historical ones. This technique makes the template pool adapt to the dynamic scene efficiently. As in the case of reliability preservation, this method also calculates a similarity measure between each target sample and the rest of the samples of the template pool; but unlike the case of reliability preservation, the target sample $\bar{n}^{f}$ need to be removed is selected by

$$\bar{n}^{f} = \arg \max_{k} \exp\left(-\frac{1}{2\lambda}\left\| n_{k}^{f} - \left(s_{\text{III}}^{f}\right)^{*}(:,:,K)\right\|^{2}\right) \tag{37}$$

.

3.7 The Fusion of two Motion cues

We have found the weight between each individual and its neighbors and also the feature vector of the motion energy for each individual in a frame using structural contextual descriptor (SCD) and saliency motion descriptor (SMD). These weights and features obtained from the two methods are fused together so that a better motion context is available which further helps in boosting the performance of anomaly detection. The results obtained after fusion has been discussed in Chapter 4 and also compared with previous anomaly detection techniques.

Let us consider $W_{FUSED} = \{\mathbf{W}_k\}_{k=1}^{M+1}, \mathbf{W}_k \in \mathbb{R}^{1\times M}$ is the fused weight vector of the k^{th} individual and $F_{FUSED} = \{\mathbf{F}_k\}_{k=1}^{M+1}, \mathbf{F}_k \in \mathbb{R}^{4\times M}$ is the corresponding fused feature vector of the *M* neighbors, whose column elements are max, min, mean and variance of the motion energyin the individual bounding box. The fused behavior difference of all the individuals with their neighbors are calculated and is denoted by $\{W_{FUSED}, F_{FUSED}\}$.

Let $\overrightarrow{W_{OF}}$ and $\overrightarrow{W_{SAL}}$ be the two *multi-dimensional* weight vectors obtained from optical flow model and the saliency model respectively. The dimension of these vectors depends on the number of neighbors. Figure 11 shows the two-dimensional representation of the weights, i.e. if each individual has two neighbors. $\overrightarrow{X}$ is the vector bisecting the angle between these two weight vectors and is computed as:

$$\overrightarrow{X} = \frac{1}{2}\left(\overrightarrow{W_{OF}} + \overrightarrow{W_{SAL}}\right) \tag{38}$$

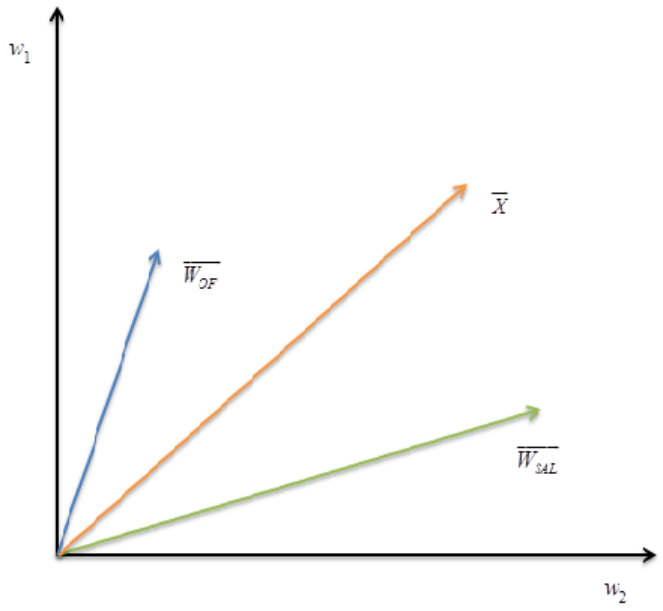


Figure 11: A vector bisecting the two weight vectors.

The unit vector along the direction of $\overrightarrow{X}$ is defined as:

$$\hat{x} = \frac{\overrightarrow{X}}{\left\|\overrightarrow{X}\right\|} \tag{39}$$

The fusion weight vector is mathematically computed as:

$$\overrightarrow{W}_{FUSED} = \left[\left(\overrightarrow{W_{OF}} \bullet \overrightarrow{X}\right)\hat{x} + \left(\overrightarrow{W_{SAL}} \bullet \overrightarrow{X}\right)\hat{x}\right] \tag{40}$$

$$\Rightarrow \overrightarrow{W}_{FUSED} = \left[\left(\overrightarrow{W_{OF}} + \overrightarrow{W_{SAL}}\right) \bullet \overrightarrow{X}\right] \hat{x} \tag{41}$$

$$\Rightarrow \vec{W}_{FUSED} = 2\left[\vec{X} \bullet \vec{X}\right] \hat{x} \tag{42}$$

$$\Rightarrow \vec{W}_{FUSED} = 2 \left\|\vec{X}\right\|^2 \hat{x} \tag{43}$$

$$\Rightarrow \vec{W}_{FUSED} = 2 \left\|\vec{X}\right\|^2 \frac{\vec{X}}{\left\|\vec{X}\right\|} \tag{44}$$

$$\Rightarrow \vec{W}_{FUSED} = 2 \left\|\vec{X}\right\| \vec{X} \tag{45}$$

Similarly, Let $\overrightarrow{F_{OF}}$ and $\overrightarrow{F_{SAL}}$ be the two *multi-dimensional* weight vectors obtained from optical flow model and the saliency model respectively. The dimension of these vectors depends on the number of neighbors. $\vec{Y}$ is the vector bisecting the angle between these two weight vectors and is computed as:

$$\vec{Y} = \frac{1}{2}\left(\overrightarrow{F_{OF}} + \overrightarrow{F_{SAL}}\right) \tag{46}$$

The unit vector along the direction of $\vec{X}$ is defined as:

$$\hat{y} = \frac{\vec{Y}}{\left\|\vec{Y}\right\|} \tag{47}$$

Using the same mathematics used for fused wrights, the fused feature vector is given as:

$$\Rightarrow \vec{F}_{FUSED} = 2 \left\|\vec{Y}\right\| \vec{Y} \tag{48}$$

Thus we have the fused weight and feature vector of all the individuals in a frame. Then for the observers found through 3D-DCT model, the temporal SCD and SMD variation is computed for each frame and this is done using Earth Mover's Distance discussed in the next section.

3.8 Earth Mover's Distance

Earth mover's distance (EMD) is defined as the measure of the distance between two probability distributions over a region D. EMD is equivalent to the 1st Mallows distance or 1st Wasserstein distance between two distributions [28]-[29] those have same integral values, as in the case of normalized histograms or probability density functions.

The *ground distance* is the distance measure between single features in a feature space. The EMD measures the dissimilarity between two multi-dimensional distributions in some feature space where the ground distance is given. This distance from the individual features are lifted to full distributions by the EMD. A set of clusters can represent a distribution where each of the clusters is represented by its mean (or mode) and by the part of the distribution that belongs to that cluster.This representation is called the *signature* of the distribution. The size of the two signatures can be of different size.

The computation of EMD is based on a solution to the well-known *transportation problem* [27]. Suppose there are several suppliers and several consumers. It is required for the suppliers, with a given amount of goods to supply the consumers, each with a given limited capacity. The cost of transporting a single unit of goods is given for each supplier-consumer pair. Then the transportation problem is defined as to find a least-

expensive flow of goods to the consumers from the suppliers satisfying the demands of the consumers. As an analogy, the consumers and the suppliers can be thought of as two different signatures. The ground distance between these two signatures can be thought of as the cost for a supplier-consumer pair. Then the problem of matching signatures can be naturally thought of as a transportation problem. The least amount of "work" required to transform one signature into the other is then intuitively the solution of the transportation problem.

The above concept can be formalized as the following linear programming problem: Let $P = \left\{ \left(p_1, w_{p_1} \right), \left(p_2, w_{p_2} \right), \left(p_3, w_{p_3} \right), ..., \left(p_m, w_{p_m} \right) \right\}$ be the first signature having *m* clusters, where p_i is the cluster representative and w_{p_i} is the weight of the cluster; and $Q = \left\{ \left(q_1, w_{q_1} \right), \left(q_2, w_{q_2} \right), \left(q_3, w_{q_3} \right), ..., \left(q_n, w_{q_n} \right) \right\}$ is the second signature with *n* clusters, where q_j is the cluster representative and w_{q_j} is the weight of the cluster ; and $\mathbf{D} = \left[d_{ij} \right]$ is the ground distance matrix where d_{ij} is the ground distance between clusters p_i and q_j .

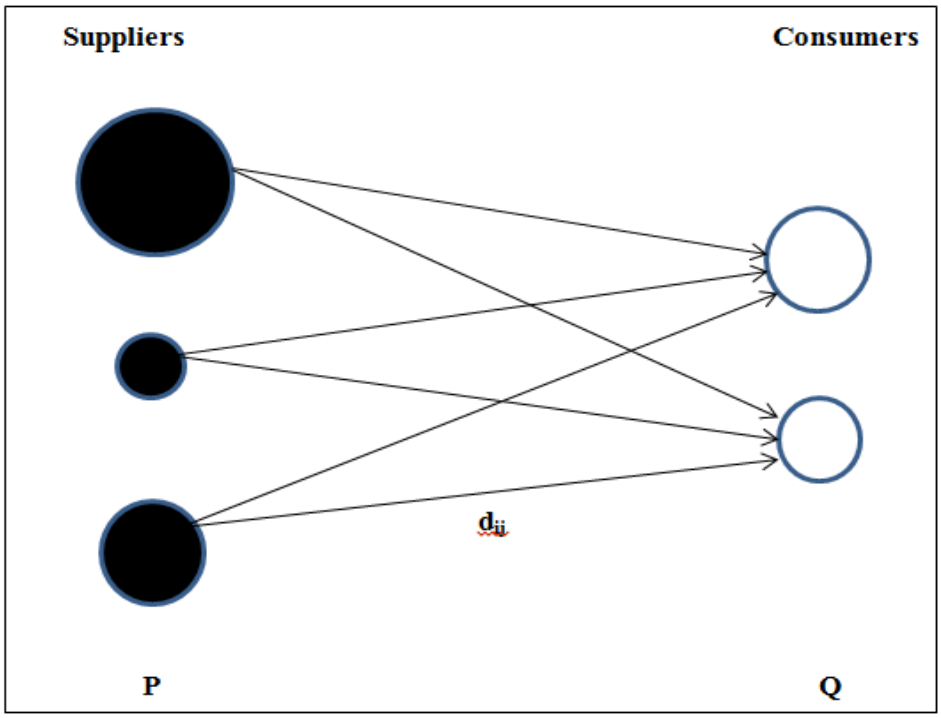


Figure 12: An example of a transportation problem with three suppliers and two consumers.

The objective function is to find a flow $F = f_{ij}$ where f_{ij} is the flow between p_i and q_j, that minimizes the overall cost given by:

$$WORK(P,Q,\mathbf{F}) = \sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij}d_{ij} \tag{49}$$

subject to the following constraints:

$$f_{ij} \geq 0 \qquad 1 \leq i \leq m, \quad 1 \leq j \leq n \tag{50}$$

$$\sum_{j=1}^{n} f_{ij} \le w_{p_i} \qquad 1 \le i \le m$$

$$\sum_{i=1}^{m} f_{ij} \le w_{q_j} \qquad 1 \le j \le n$$

$$\sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij} = \min\left(\sum_{i=1}^{m} w_{p_i}, \sum_{j=1}^{n} w_{q_j}\right),$$

The first constraint takes care of the fact that the "supplies" are being moved from P to Q and not vice-versa. The second constraint takes into account that the amount of supplies sent by the clusters in P does not exceed their weights. The third constraint takes into account that the amount of supplies received by the clusters in Q is no more than their weights. The last constraint takes care of the fact that the maximum amount of supplies possible be moved from P to Q. This amount is called the *total flow*. As soon as the transportation problem is solved, and we have found the optimal flow F, the earth mover's distance is defined as the work normalized by the total flow:

$$EMD(P,Q) = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij} d_{ij}}{\sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij}} \tag{51}$$

The normalization factor helps to avoid giving much importance to smaller signatures in the case of partial matching.

EMD is useful in analyzing similarity of two distributions which may have different dimensions. Hence, based on the foundation the Earth mover's distance is found useful in detecting any frame level anomaly and it is explained below.

3.8.1 Frame Level Anomaly Detection using EMD:

The corresponding observers found from 3D-DCT model are compared between frames to get the frame-level abnormality. The abnormal frames are labelled as output. Let the number of observers be B. The contextual descriptor of the n^{th} $\left(n=1,2,3,...,B\right)$ observer at t^{th} time be $\left\{\mathbf{W}_t^n,\mathbf{F}_t^n\right\}$ and at $\left(t+1\right)^{th}$ time be $\left\{\mathbf{W}_{t+1}^n,\mathbf{F}_{t+1}^n\right\}$. The contextual variation of the observers is computed by the EMD [28].

The difference between the adjacent frames is given by

$$d_n^{t+1}=1-EMD\left(\mathbf{W}_t^n,\mathbf{W}_{t+1}^n,\mathbf{F}_t^n,\mathbf{F}_{t+1}^n\right) \tag{52}$$

$$AD_{frame}^{t+1}=1-\left(\sum_{n=1}^{B}d_n^{t+1}\right)\Big/B \tag{53}$$

Finally, the anomalous frames are declared those satisfy the average value $AD_{frame}^{t+1}>0.5$. The threshold for detecting an anomalous frame is set as 0.5 which is a reasonable choice in concord with the experiments done.

Table 2: Algorithm of the Proposed Model

ALGORITHM

Setting the Parameters

Input: Video Sequence

Method:

1. Pedestrian Detection, i.e. detect each individual in a frame.
2. SCD computation of each pedestrian in all the frames.
3. SMD computation of each pedestrian in all the frames.
4. Multiple objects tracking to seek B observers.
5. Fusion of the two motion cues obtained from SCD and SMD variation of the observers.
6. Computing the fused motion variation d_n^{t+1} of the n^{th} observer, and obtain the frame-level abnormal degree AD_{frame}^{t+1}.

Output: AD_{frame}^{t+1}, the frame-level abnormality is decided by $AD_{frame}^{t+1} > 0.5$.

References

[1] Wu, Shandong, Brian E. Moore, and Mubarak Shah. "Chaotic invariants of lagrangian particle trajectories for anomaly detection in crowded scenes." In*Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pp. 2054-2060. IEEE, 2010.

[2] Dollár, Piotr, Christian Wojek, Bernt Schiele, and Pietro Perona. "Pedestrian detection: A benchmark." In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pp. 304-311. IEEE, 2009.

[3] Tsukiyama, Toshifumi, and Yoshiaki Shirai. "Detection of the movements of persons from a sparse sequence of tv images." *Pattern Recognition* 18, no. 3 (1985): 207-213.

[4] Gavrila, Dariu M., and Vasanth Philomin. "Real-time object detection for "smart" vehicles." In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, vol. 1, pp. 87-93. IEEE, 1999.

[5] Song, Yang, Xiaolin Feng, and Pietro Perona. "Towards detection of human motion." In *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, vol. 1, pp. 810-817. IEEE, 2000.

[6] Papageorgiou, Constantine, and Tomaso Poggio. "A trainable system for object detection." *International Journal of Computer Vision* 38, no. 1 (2000): 15-33.

[7] Cong, Yang, Junsong Yuan, and Ji Liu. "Abnormal event detection in crowded scenes using sparse representation." *Pattern Recognition* 46, no. 7 (2013): 1851-1864.

[8] X. Cui, Q. Liu, M. Gao, and D. Metaxas, "Abnormal detection using interaction energy potentials," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Providence, RI, USA, 2011, pp. 3161–3167.

[9] Cheng, Hsu-Yung, and Jenq-Neng Hwang. "Integrated video object tracking with applications in trajectory-based event detection." *Journal of Visual Communication and Image Representation* 22, no. 7 (2011): 673-685.

[10] Dalal, Navneet, and Bill Triggs. "Histograms of oriented gradients for human detection." In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1, pp. 886-893. IEEE, 2005.

[11] Dollár, Piotr, Boris Babenko, Serge Belongie, Pietro Perona, and Zhuowen Tu. "Multiple component learning for object detection." *Computer Vision–ECCV*

2008 (2008): 211-224.

[12] Ess, Andreas, Bastian Leibe, Konrad Schindler, and Luc Van Gool. "A mobile vision system for robust multi-person tracking." In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pp. 1-8. IEEE, 2008.

[13] Felzenszwalb, Pedro, David McAllester, and Deva Ramanan. "A discriminatively trained, multiscale, deformable part model." In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pp. 1-8. IEEE, 2008.

[14] Leibe, Bastian, Nico Cornelis, Kurt Cornelis, and Luc Van Gool. "Dynamic 3d scene analysis from a moving vehicle." In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pp. 1-8. IEEE, 2007.

[15] Saligrama, Venkatesh, and Zhu Chen. "Video anomaly detection based on local statistical aggregates." In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pp. 2112-2119. IEEE, 2012.

[16] W. Ge, R. Collins, and R. Ruback, "Vision-based analysis of small groups in pedestrain crowds," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 5, pp. 1003–1016, May 2012.

[17] Xiang, Tao, and Shaogang Gong. "Incremental and adaptive abnormal behaviour detection." *Computer Vision and Image Understanding* 111, no. 1 (2008): 59-73.

[18] Zhang, Yanhao, Lei Qin, Hongxun Yao, and Qingming Huang. "Abnormal crowd behavior detection based on social attribute-aware force model." In*Image Processing (ICIP), 2012 19th IEEE International Conference on*, pp. 2689-2692. IEEE, 2012.

[19] Varadarajan, Jagannadan, and Jean-Marc Odobez. "Topic models for scene analysis and abnormality detection." In *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*, pp. 1338-1345. IEEE, 2009.

[20] Wu, Si, Hau-San Wong, and Zhiwen Yu. "A bayesian model for crowd escape behavior detection." *Circuits and Systems for Video Technology, IEEE Transactions on* 24, no. 1 (2014): 85-98.

[21] Anjum, Nadeem, and Andrea Cavallaro. "Multifeature object trajectory clustering for video analysis." *Circuits and Systems for Video Technology, IEEE Transactions on* 18, no. 11 (2008): 1555-1564.

[22] Piciarelli, Claudio, Christian Micheloni, and Gian Luca Foresti. "Trajectory-based

anomalous event detection." *Circuits and Systems for Video Technology, IEEE Transactions on* 18, no. 11 (2008): 1544-1554.

[23] Kratz, Louis, and Ko Nishino. "Anomaly detection in extremely crowded scenes using spatio-temporal motion pattern models." In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pp. 1446-1453. IEEE, 2009.

[24] Chang, C. C., and C. J. Lin. "http:// www. csie. ntu. edu. tw/~ cjlin/ libsvm."*LIBSVM: a library for support vector machines* (2001).

[25] Cartwright, Dorwin Ed, and Alvin Ed Zander. "Group dynamics research and theory." (1953).

[26] Granovetter, Mark. "Threshold models of collective behavior." *American journal of sociology* (1978): 1420-1443.

[27] Hitchcock, Frank L. "The distribution of a product from several sources to numerous localities." *Journal of mathematics and physics* 20, no. 1 (1941): 224-230.

[28] Levina, Elizaveta, and Peter Bickel. "The earth mover's distance is the Mallows distance: some insights from statistics." In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, vol. 2, pp. 251-256. IEEE, 2001.

[29] Mallows, C. L. "A note on asymptotic joint normality." *The Annals of Mathematical Statistics* (1972): 508-515.

[30] Li, Xi, Anthony Dick, Chunhua Shen, Anton Van Den Hengel, and Hanzi Wang. "Incremental learning of 3D-DCT compact representations for robust visual tracking." *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 35, no. 4 (2013): 863-881.

[31] Berclaz, Jerome, Francois Fleuret, Engin Türetken, and Pascal Fua. "Multiple object tracking using k-shortest paths optimization." *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 33, no. 9 (2011): 1806-1819.

[32] Huang, Chang, Yuan Li, and Ramakant Nevatia. "Multiple target tracking by learning-based hierarchical association of detection responses." *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 35, no. 4 (2013): 898-910.

[33] Yuan, Yuan, Jianwu Fang, and Qi Wang. "Online anomaly detection in crowd scenes via structure analysis." *Cybernetics, IEEE Transactions on* 45, no. 3 (2015): 548-561.

[34] Bourbaki, Nicolas. *Algebra I: chapters 1-3*. Springer Science & Business Media, 1998.

[35] Zhang, Yanhao, Lei Qin, Hongxun Yao, Pengfei Xu, and Qingming Huang. "Beyond particle flow: Bag of trajectory graphs for dense crowd event recognition." In *Image Processing (ICIP), 2013 20th IEEE International Conference on*, pp. 3572-3576. IEEE, 2013.

[36] Lowe, David G. "Distinctive image features from scale-invariant keypoints."*International journal of computer vision* 60, no. 2 (2004): 91-110.

[37] Kim, Jaechul, and Kristen Grauman. "Observe locally, infer globally: a space-time MRF for detecting abnormal activities with incremental updates." In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pp. 2921-2928. IEEE, 2009.

[38] Mahadevan, Vijay, Weixin Li, Viral Bhalodia, and Nuno Vasconcelos. "Anomaly detection in crowded scenes." (2010): 1975-1981.

[39] W. Li, V. Mahadevan, and N. Vasconcelos, "Anomaly detection and localization in crowded scenes," *IEEE Trans. Pattern Anal. Mach. Intell.*,vol. 36, no. 1, pp. 18–32, Jan. 2014.

[40] Jing, Huiyun, Xin He, Qi Han, Ahmed A. Abd El-Latif, and Xiamu Niu. "Saliency detection based on integrated features." *Neurocomputing* 129 (2014): 114-121.

[41] Kim, Wonjun, and Jae-Joon Han. "Video saliency detection using contrast of spatiotemporal directional coherence." *Signal Processing Letters, IEEE* 21, no. 10 (2014): 1250-1254.

[42] Liu, Zhi, Xiang Zhang, Shuhua Luo, and Olivier Le Meur. "Superpixel-based spatiotemporal saliency detection." *Circuits and Systems for Video Technology, IEEE Transactions on* 24, no. 9 (2014): 1522-1540.

[43] Zhang, Qiang, Yueling Chen, and Long Wang. "Multisensor video fusion based on spatial–temporal salience detection." *Signal Processing* 93, no. 9 (2013): 2485-2499.

[44] Koutra, Danai, Ankur Parikh, Aaditya Ramdas, and Jing Xiang. *Algorithms for graph similarity and subgraph matching*. Technical Report of Carnegie-Mellon-University, 2011.

[45] Gangapure, Vijay N., Susmit Nanda, Ananda S. Chowdhury, and Xiaoyi Jiang. "Causal video segmentation using superseeds and graph matching." In *Graph-Based*

Representations in Pattern Recognition, pp. 282-291. Springer International Publishing, 2015.

[46] Chaudhry, Rizwan, Arunkumar Ravichandran, Georg Hager, and René Vidal. "Histograms of oriented optical flow and binet-cauchy kernels on nonlinear dynamical systems for the recognition of human actions." In *Computer Vision and Pattern Recog*

[47] Liu, Ce. "Beyond pixels: exploring *nition, 2009. CVPR 2009. IEEE Conference on*, pp. 1932-1939. IEEE, 2009. new representations and applications for motion analysis." PhD diss., Massachusetts Institute of Technology, 2009.

[48] Shao, Ling, Ling Ji, Yan Liu, and Jianguo Zhang. "Human action segmentation and recognition via motion and shape analysis." *Pattern Recognition Letters* 33, no. 4 (2012): 438-445.

[49] Mehran, Ramin, Brian E. Moore, and Mubarak Shah. "A streakline representation of flow in crowded scenes." In *Computer Vision–ECCV 2010*, pp. 439-452. Springer Berlin Heidelberg, 2010.

[50] Mehran, Ramin, Akira Oyama, and Mubarak Shah. "Abnormal crowd behavior detection using social force model." In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pp. 935-942. IEEE, 2009.

[51] Thida, Myo, How-Lung Eng, and Paolo Remagnino. "Laplacian eigenmap with temporal constraints for local abnormality detection in crowded scenes."*Cybernetics, IEEE Transactions on* 43, no. 6 (2013): 2147-2156.

[52] Benezeth, Yannick, Pierre-Marc Jodoin, Venkatesh Saligrama, and Christophe Rosenberger. "Abnormal events detection based on spatio-temporal co-occurences." In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pp. 2458-2465. IEEE, 2009.

CHAPTER 4

EXPERIMENTAL RESULTS

This chapter deals with the experimental setup and results. At first, the dataset is illustrated in Section 4.1. Parameter setting and evaluation criterions are explained in Section 4.2 and finally Section 4.3 deals with all the results obtained at each processing step and also the effectiveness of the model is shown in ROC.

4.1 Datasets

The performance of the proposed method is evaluated on the publicly available USCD dataset and is explained as follows:

USCD Dataset: The USCD dataset [7] is useful to test the ability of the proposed model to detect any local abnormality. The local abnormalities are illustrated as: (a) irregular behaviors in the surroundings (e.g., people cycling or skating across walkways) and (b) unusual individuals in crowd (e.g. individuals on wheel chair). The dataset contains two different scenarios viz. ped1 and ped2. There are 34 normal video sequences for training and 36 abnormal video sequences for testing in ped1 video set. In the case of ped2 video set, there are 16 normal video sequences for training and 14 abnormal video sequences for testing. There are 200 video frames for each of the video sequences in ped1 and the resolution of each image is 158 x 238 and 180 video frames for each of the video sequences in ped2 and the resolution of each image is 360 x 240. In this dataset, the optimal ξ for this dataset is learned to be 1, and it indicates that the magnitude plays more important role in ped1 dataset. In the ped2 dataset, it also depends on the magnitude inconsistency but the learned ξ is 0.9. But this goes against to the concept of SHOF that whenever ξ is large, the main attribute for anomaly detection is motion direction. However, from the observation on the first frames of video sequences in ped2, all the individuals move at the same direction in the normal frames, and the motion difference is consistent irrespective of the value of ξ, i.e. the experimental results of ped2 dataset is not affected by the value of ξ.

4.2 Experimentation Details

This section deals with the details of the basic parameter settings and the criterion for evaluation of the proposed model. The specification of the machine used and the computing platform has also been stated.

4.2.1 Parameters

The algorithm described in Chapter 3 requires a few parameters to be set. The first parameter is the size of the template pool *K*. Considering the computational efficiency and robustness of the 3D-DCT based multi object tracker, K is empirically set as 6. The second parameter is the constants in equation (6). The constants *a*, *b*, *m* and *n* in equation (6) are set to 1, 1, 3 and 1 respectively [1]. The parameter '*indx'* represent the number of initial frames considered normal and is set to 5. These frames are used for learning the optimal ξ. The number of nearest neighbors surrounding an examined target, *k_NN* is set to 5 experimentally. Another important parameter is the number of superpixels, *k* used as an input to SLIC. This value is chosen to be 1700 to accurately detect temporal saliency. The threshold parameter $T_1$for SDFD to determine the superpixels in motion is set to be 0.15 and the threshold parameter T_2 used to obtain the temporal saliency map is set to be 0.10.

The experiments are performed on a PC with Intel(R) Core(TM) i5-2400 processor having 3.10 GHz speed and 8 GB RAM. All the simulations required for the experiment is computed in MATLAB 2015a.

4.2.2 Criterions for Evaluation

The efficiency pf the proposed method is evaluated by two criterions. The first of them is the receiver operating characteristic (ROC) and the second one is the area under ROC (AUC). ROC reflects the relationship between *sensitivity* and *specificity.* Sensitivity or true positive rate (TPR) is defined as the rate of correctly labeled frames and Specificity or false positive rate (FPR) is defined as the rate of incorrectly labeled

where, TPR is defined as: $$TPR = \frac{TruePositive}{TruePositive + FalseNegative}.$$

FPR is defined as: $$FPR = \frac{FalsePositive}{TrueNegative + FalsePositive}.$$

The terms true positive, true negative, false positive and false negative are illustrated in Table 3 below, where ground truth corresponds to the known label of the frames and test data is the value obtained from the experiment.

Table 3: Confusion Matrix

TEST / GROUND TRUTH	Positive	Negative
Positive	**TRUE POSITIVE**	**FALSE NEGATIVE**
Negative	**FALSE POSITIVE**	**TRUE NEGATIVE**

4.3 Results

This section illustrates the results obtained at each step of the proposed method and finally depicts the ROC curve for comparison with other methods.

4.3.1 Experiment 1: Pedestrian Detection

The number of histogram channels is taken as 9 for the orientation binning process of HOG feature descriptor. The block geometry used is rectangular R-HOG blocks, where each cell has 8 x 8 pixels, 4 such cells constitute a block i.e. each block has 16 x 16 pixels and 50 percent overlap of the blocks is considered; and L2-norm is used to normalize the feature descriptor vector. And finally a publicly available Matlab verison of SVM [2] is used to detect the pedestrians.

The final output of pedestrian detection is shown in the following figures, where Figure 13shows the pedestrians detected for some of the frames of ped1 dataset and Figure 14 shows the pedestrians detected for some of the frames of ped2 dataset.

(a) Frame no. 1

(b) Frame no. 21

(c) Frame no. 41

(d) Frame no. 61

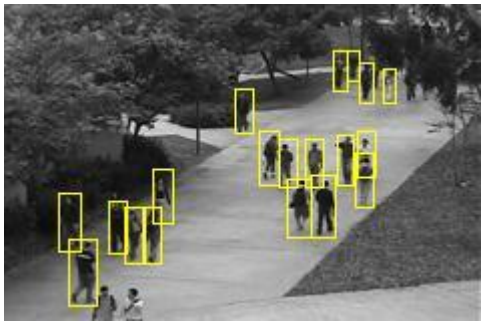
(e) Frame no. 81

(f) Frame no. 101

(g) Frame no. 121

(h) Frame no. 141

(i) Frame no. 161



(j) Frame no. 181

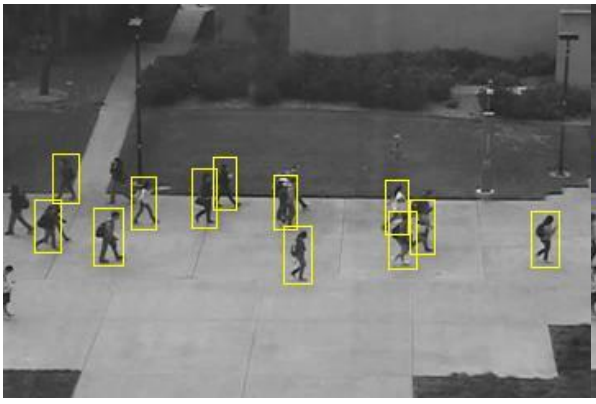
Figure 13: Pedestrians detection shown by yellow bounding boxes for 10 different frames of USCD ped1 dataset.

(a) Frame no. 10

(b) Frame no. 18

(c) Frame no. 29

(d) Frame no. 43



(e) Frame no. 65

(f) Frame no. 88

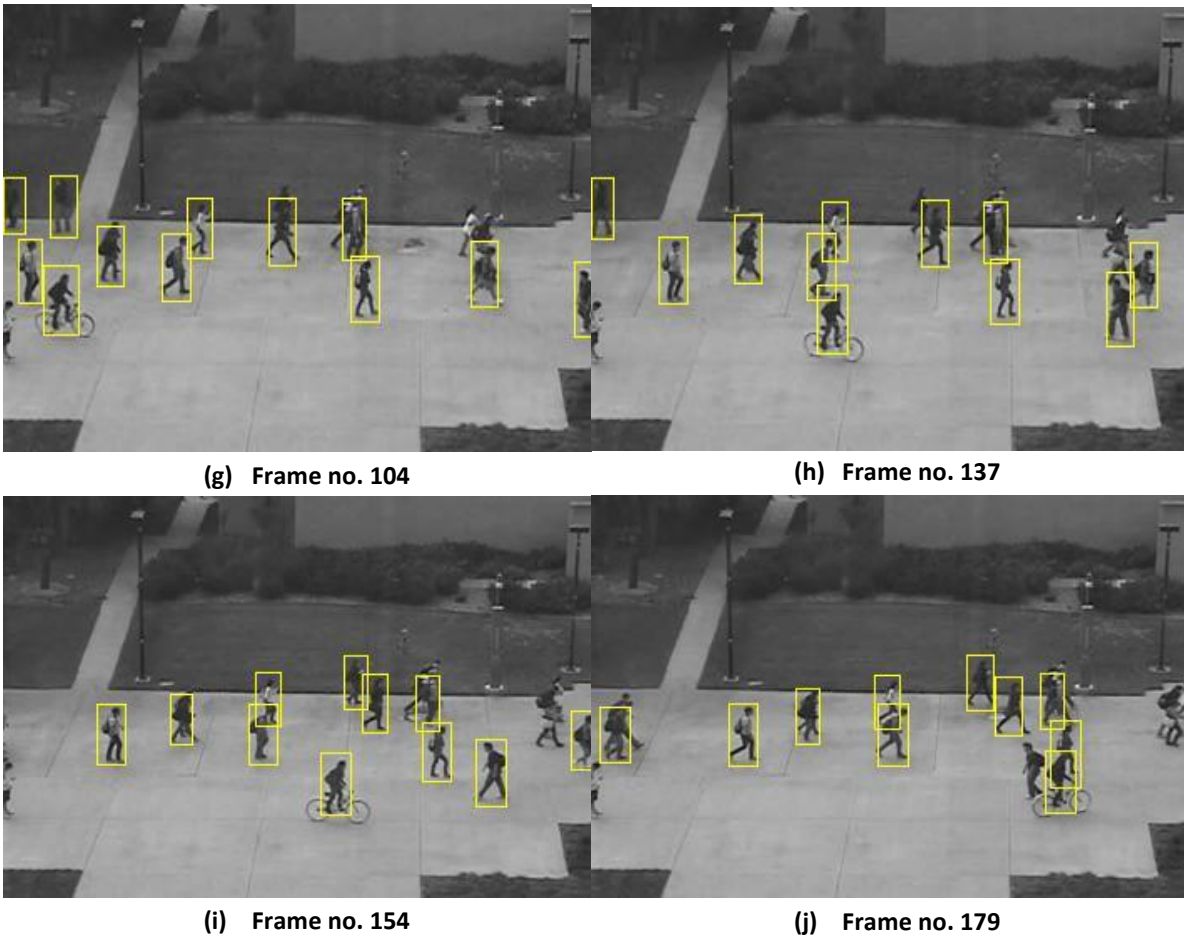
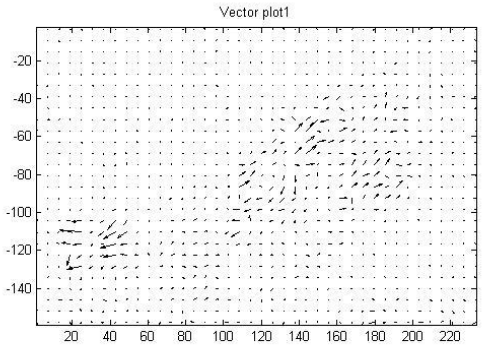


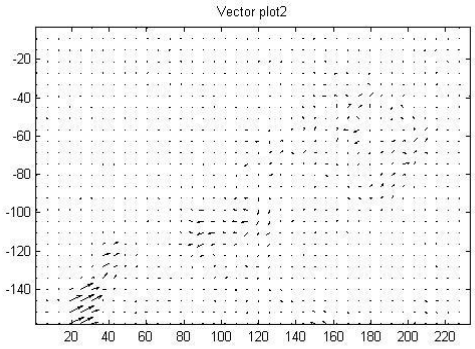
Figure 14: Pedestrians detection shown by yellow bounding boxes for 10 different frames of USCD ped2 dataset.

4.3.2 Experiment 2: Motion Context using Optical Flow

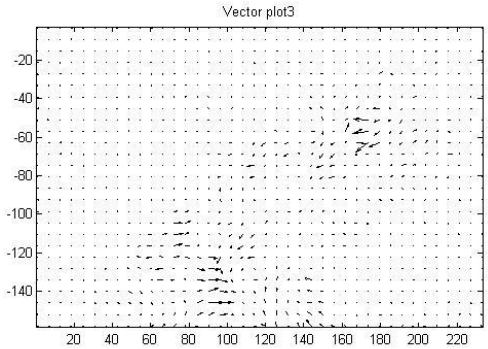
The optical flow of two consecutive frames is evaluated using [3] and the results obtained for ped1 dataset and ped2 dataset are shown below in Figure 15 and Figure 16 respectively.



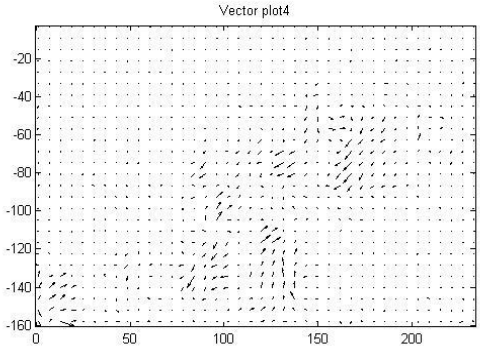
(a) Frames 1 and 2



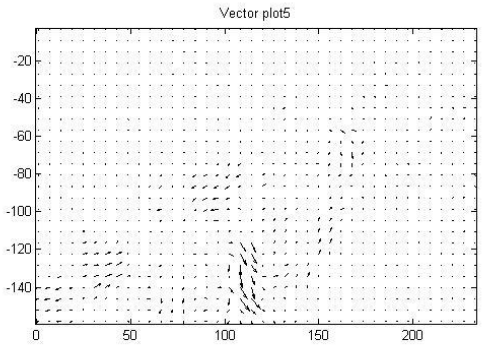
(b) Frames 45 and 46



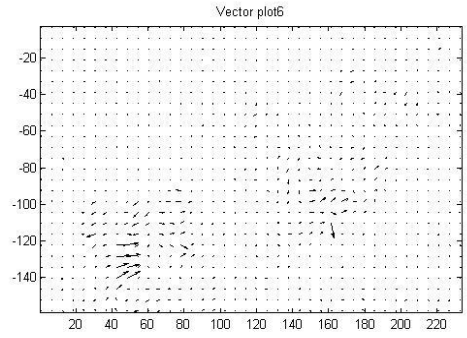
(c) Frames 78 and 79



(d) Frames 102 and 103

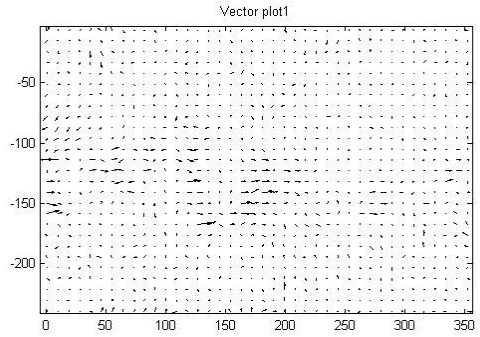


(e) Frames 139 and 140

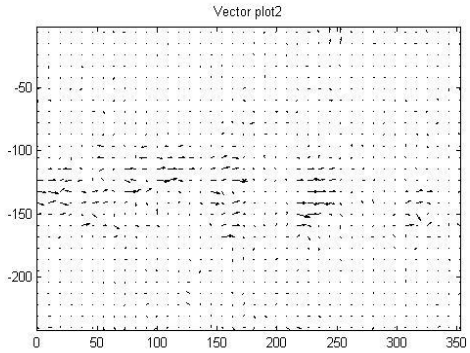


(f) Frames 183 and 184

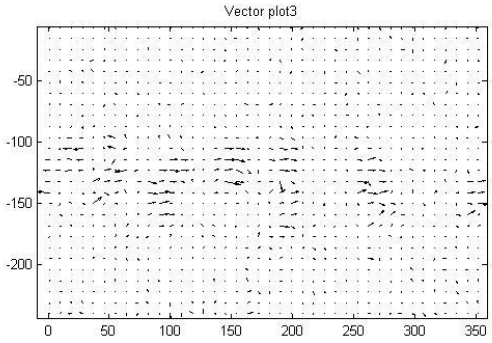
Figure 15: Optical Flow between different frames of the ped1 dataset.



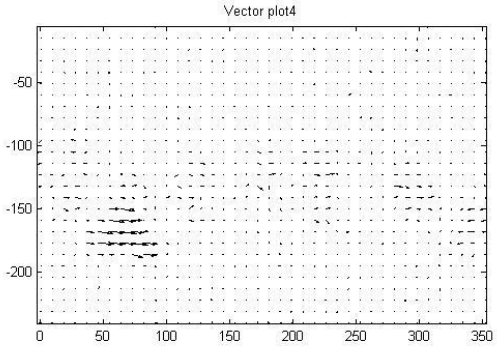
(a) Frames 1 and 2



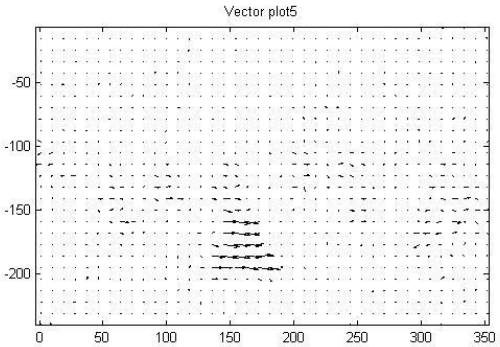
(b) Frames 45 and 46



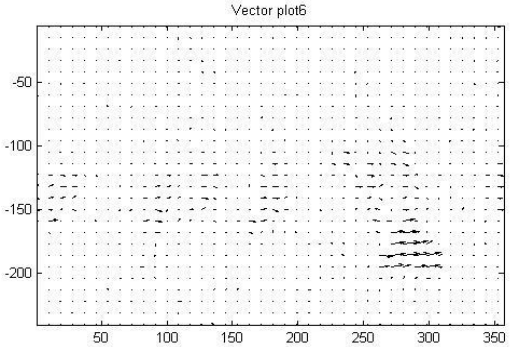
(c) Frames 82 and 83



(d) Frames 111 and 112



(e) Frames 140 and 141

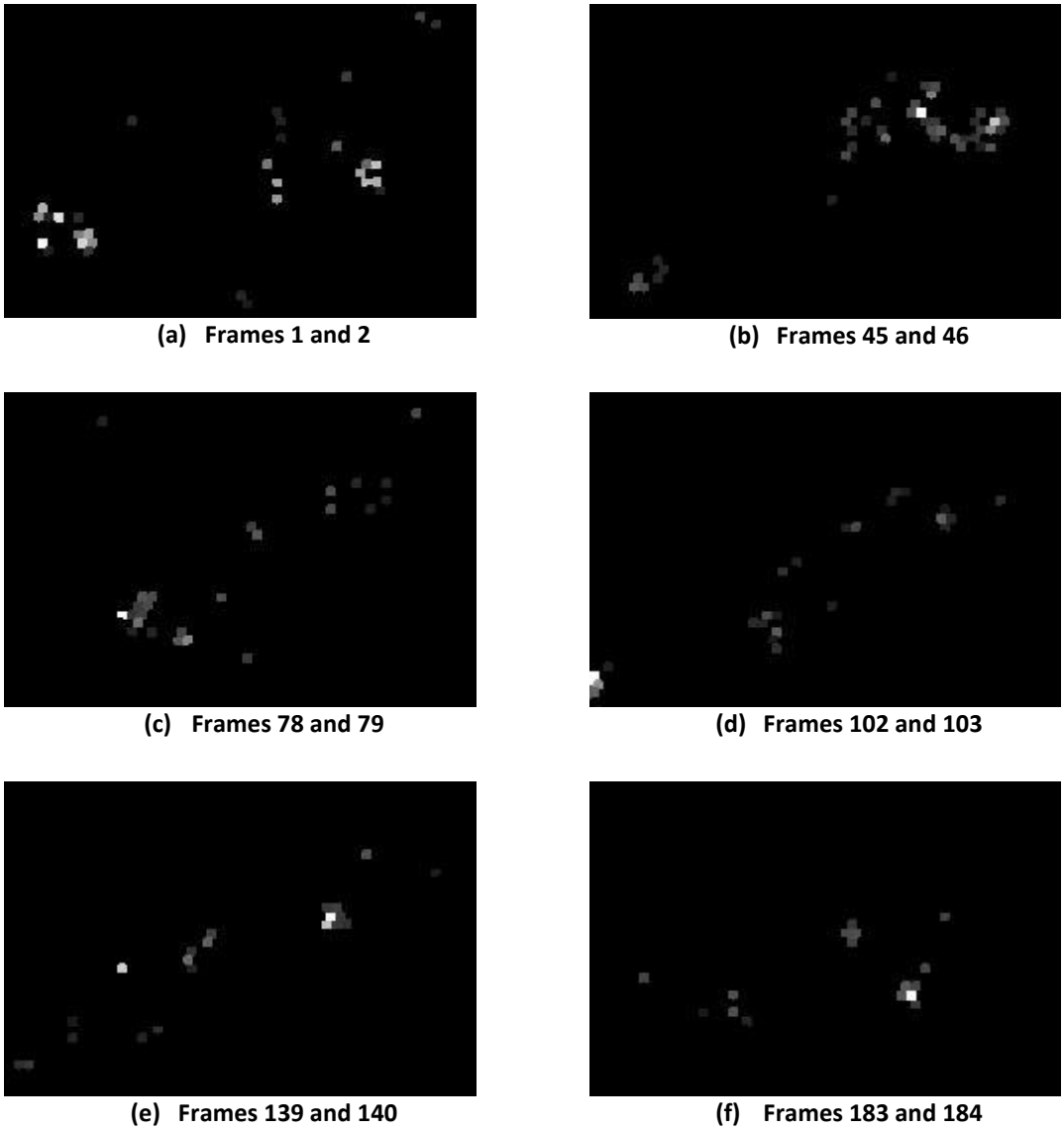


(f) Frames 178 and 179

Figure 16: Optical Flow between frames of the ped2 dataset.

4.3.3 Experiment 3: Motion Context using Saliency Map

The temporal saliency map of all the frames is obtained as shown in Figure 17 and Figure 18 for USC ped1 and ped2 datasets respectively, and is used as another cue for extracting the motion information.

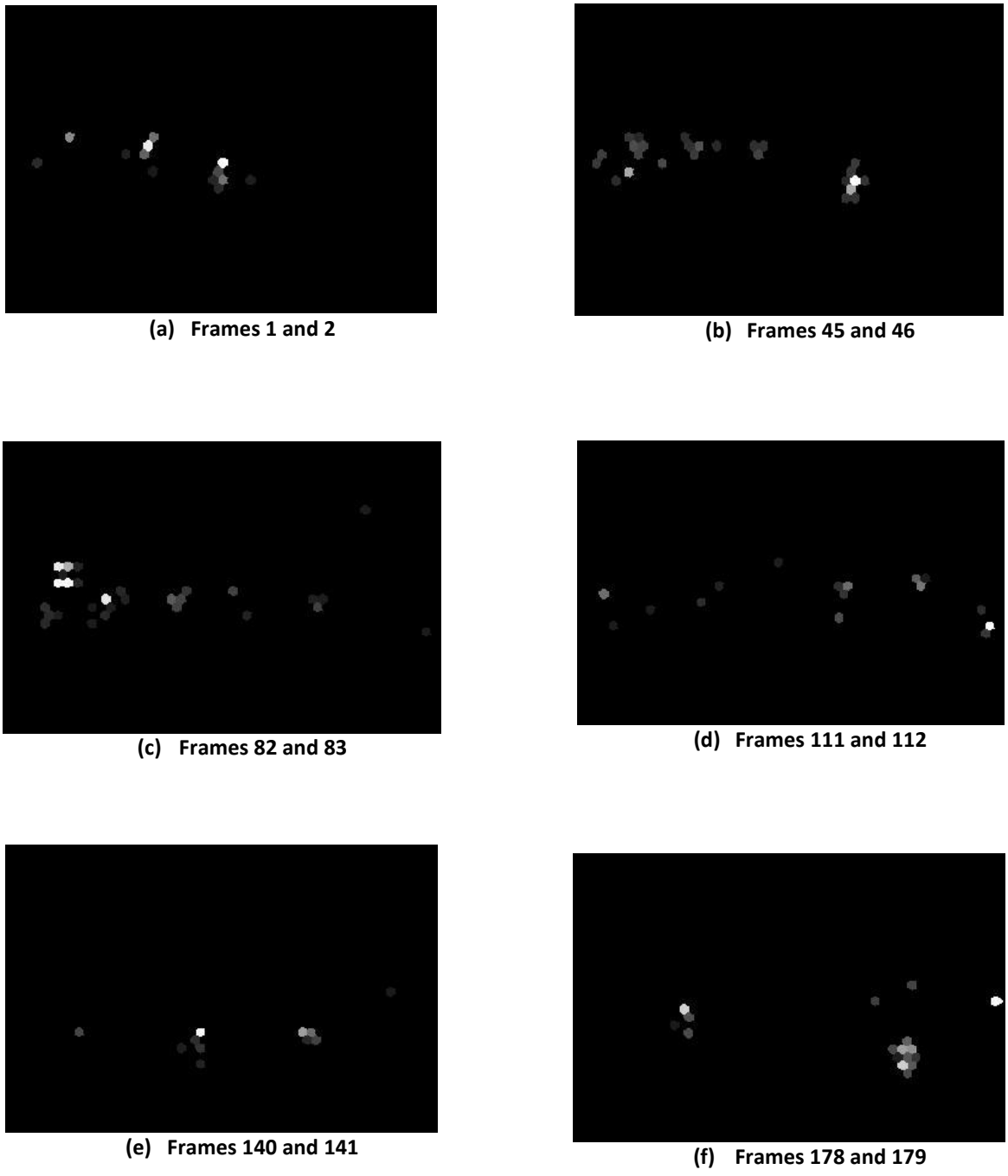


(a) Frames 1 and 2 (b) Frames 45 and 46

(c) Frames 78 and 79 (d) Frames 102 and 103

(e) Frames 139 and 140 (f) Frames 183 and 184

Figure 17: Temporal Saliency Map between frames of the ped1 dataset.



(a) Frames 1 and 2

(b) Frames 45 and 46

(c) Frames 82 and 83

(d) Frames 111 and 112

(e) Frames 140 and 141

(f) Frames 178 and 179

Figure 18: Optical Flow between frames of the ped2 dataset.

4.3.4 Experiment 5: 3 dimensional discrete cosine transforms (3D-DCT)

The 3D-DCT multi-object tracker is used here to seek the stable individuals, called *observers*. The temporal SCD and SMD variation of these observers are computed for each frame. The observers detected are marked with "red" bounding box and the normal individuals are marked with "yellow" bounding box.

The observers obtained from the ped1 dataset and ped2 dataset are shown below in Figure 19 and Figure 20 respectively.

(a) Frame no. 21

(b) Frame no. 39

(c) Frame no. 51

(d) Frame no. 75

(e) Frame no. 99

(f) Frame no. 119

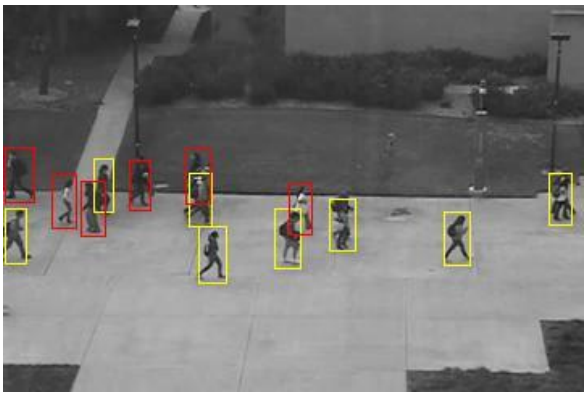
(g) Frame no. 131

(h) Frame no. 163

(i) Frame no. 179

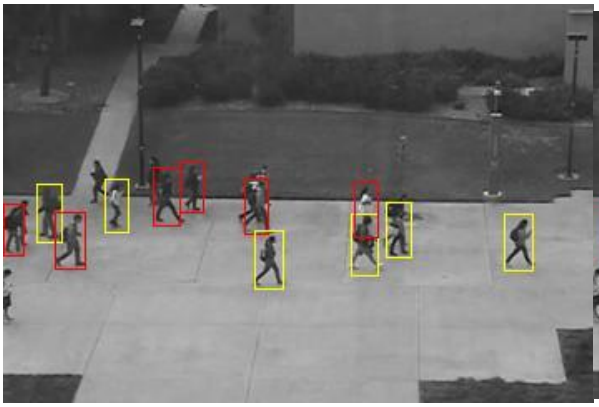
(j) Frame no. 199

Figure 19: Observers shown by red bounding boxes for 10 different frames of USCD ped1 dataset.

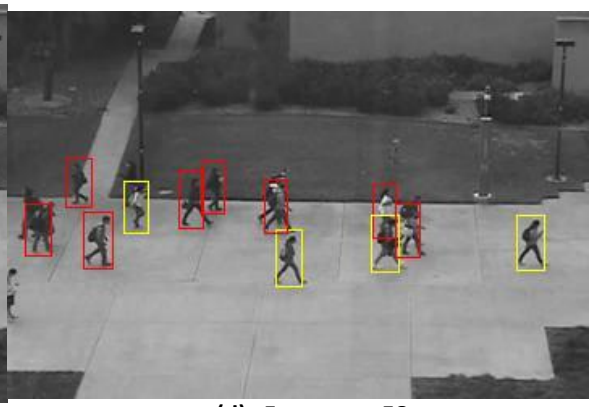


(a) Frame no. 9

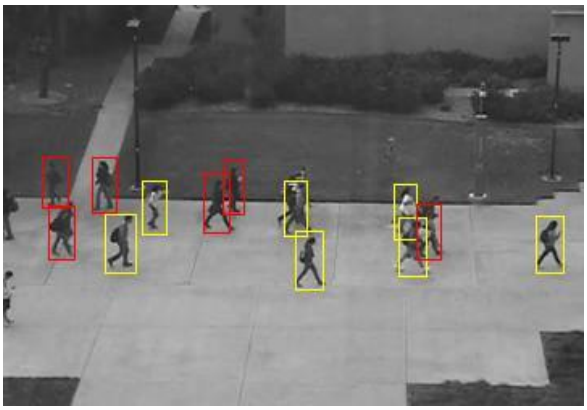
(b) Frame no. 27



(c) Frame no. 42

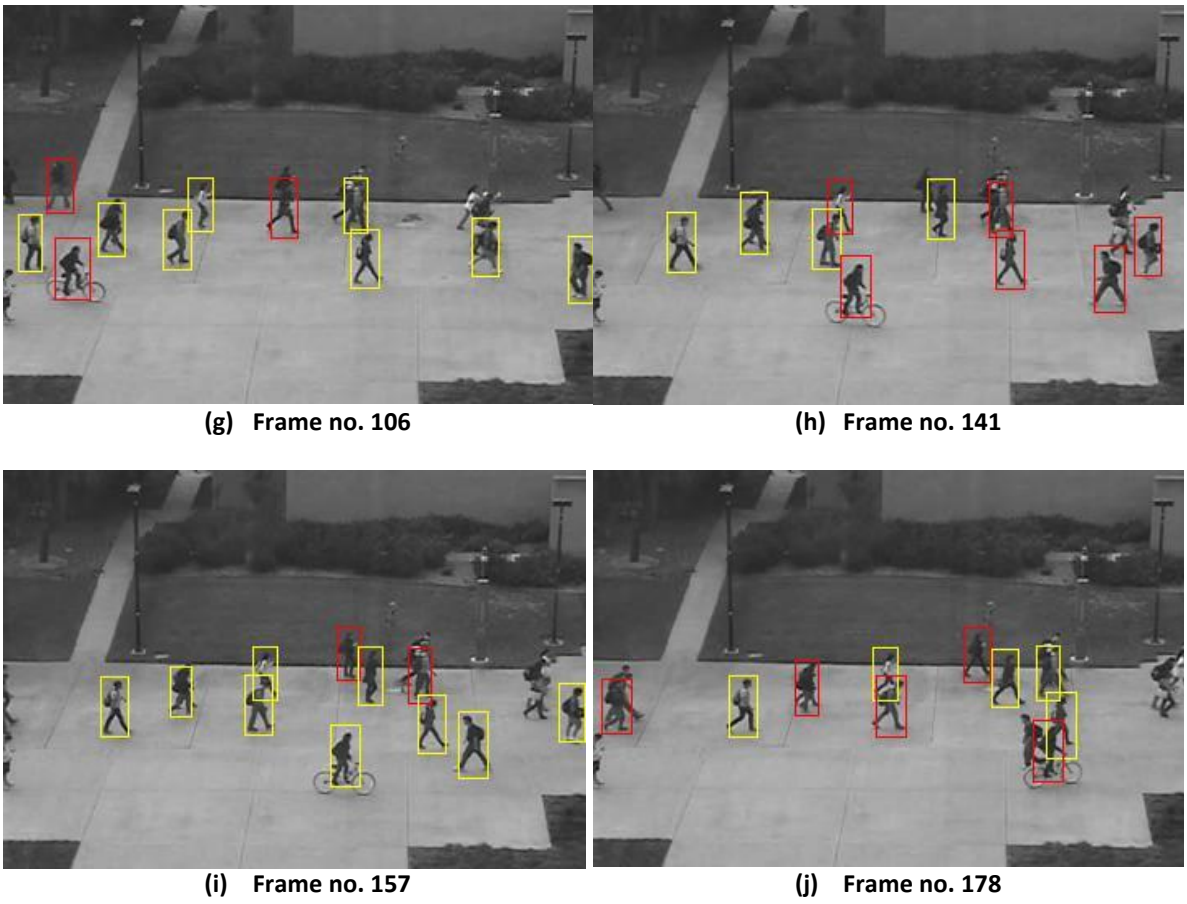


(d) Frame no. 58



(e) Frame no. 74

(f) Frame no. 91



(g) Frame no. 106 (h) Frame no. 141

(i) Frame no. 157 (j) Frame no. 178

Figure 20: Observers shown by red bounding boxes for 10 different frames of USCD ped2 dataset.

4.3.5 Receiver operating characteristic (ROC)

Figure 21and Figure 22demonstrates the frame level ROC comparisons for USCD ped1 and ped2 dataset respectively, and the AUC comparisons are shown in Table 4 below. The proposed method has superior performance as compared to the works of Adam *et. al.*[4], Kim *et. al.* [5], Mehran *et. al.*[6].

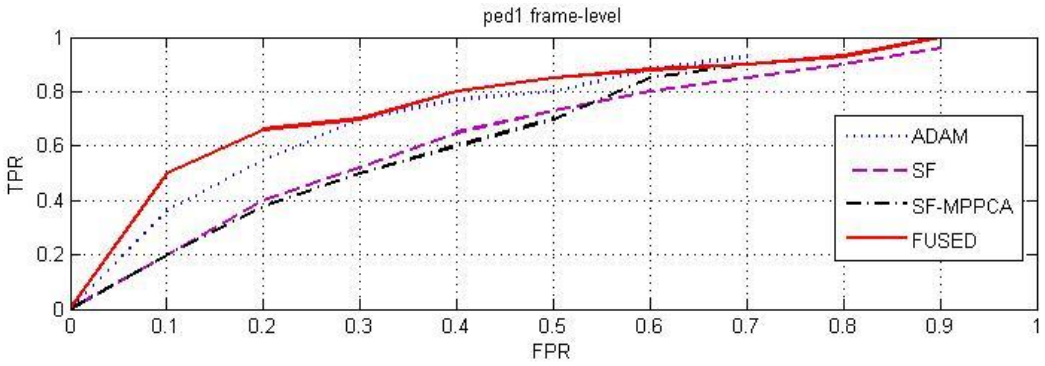


Figure 21: Frame-level ROC comparison in USCD ped2 dataset.

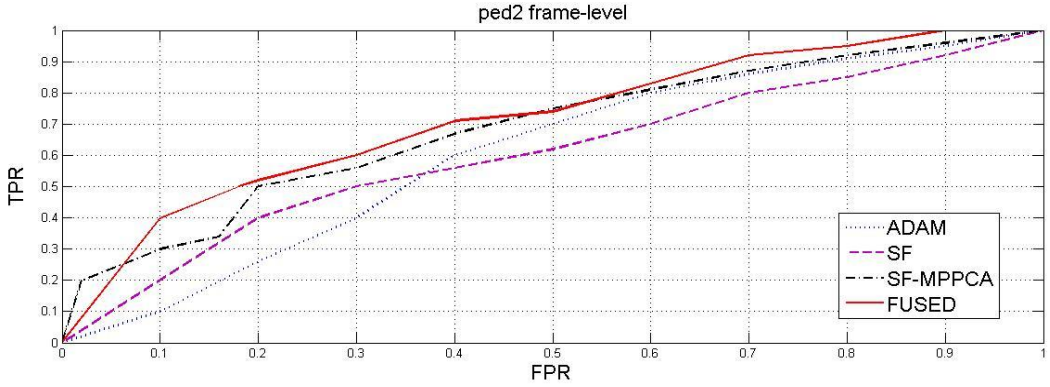


Figure 22: Frame-level ROC comparison in USCD ped2 dataset.

Table 4: Frame Level AUC Comparison for Anomaly Detection in USCD Dataset

Method	ped1	ped2
ADAM [4]	0.650	0.63
SF-MPPCA [5]	0.590	0.71
SF [6]	0.670	0.63
FUSED METHOD	0.7720	0.7172

References

[1] Yuan, Yuan, Jianwu Fang, and Qi Wang. "Online anomaly detection in crowd scenes via structure analysis." *Cybernetics, IEEE Transactions on* 45, no. 3 (2015): 548-561.

[2] Chang, C. C., and C. J. Lin. "http:// www. csie. ntu. edu. tw/~ cjlin/ libsvm."*LIBSVM: a library for support vector machines* (2001).

[3] Horn, Berthold K., and Brian G. Schunck. "Determining optical flow." In *1981 Technical symposium east*, pp. 319-331. International Society for Optics and Photonics, 1981.

[4] Adam, Amit, Ehud Rivlin, Ilan Shimshoni, and David Reinitz. "Robust real-time unusual event detection using multiple fixed-location monitors." *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 30, no. 3 (2008): 555-560.

[5] Kim, Jaechul, and Kristen Grauman. "Observe locally, infer globally: a space-time MRF for detecting abnormal activities with incremental updates." In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pp. 2921-2928. IEEE, 2009.

[6] Mehran, Ramin, Akira Oyama, and Mubarak Shah. "Abnormal crowd behavior detection using social force model." In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pp. 935-942. IEEE, 2009.

[7] (2013). *UCSD Anomaly Detection Dataset* [Online]. Available: http://www.svcl.ucsd.edu/projects/anomaly/dataset.html.

CHAPTER 5

Conclusions and Future Work

In this chapter, the thesis work has been concluded with a brief overview on key contributions of the work. Future directions related to the proposed solutions for anomaly detection are presented at the end of the chapter.

5.1 Conclusions

In today's world security has become a primary issue for any video surveillance system. The problem of anomaly detection has been a topic of interest in the recent past in the computer vision community due to its wide applications in various fields like anomalous event detection, suspicious activity detection and in many other surveillance systems. The proposed solution for detecting anomalies evolves from the motion cues obtained from two different methods namely, the optical flow concept (SCD) in pixel level approach and temporal saliency concept (SMD) in superpixel level approach. A robust multi-object tracker is then designed for target association in different frames. The tracker needs to track only the stable individuals. Finally, the SCD and SMD variation is computed to detect any crowd abnormality. From the testing results obtained from the USCD dataset, the ROC and AUC has been calculated which shows significant results.

5.2 Scope of Future Work

As a scope of future work, we will examine if any other fusion technique can make the model more robust to anomalies. However for this model to function in severe weather conditions like foggy and rainy days, one possible solution can be incorporating the multi spectral clues. Also, we will examine if superpixel extraction can be made faster which in turn would further reduce the execution time of the proposed algorithm. Finally, we look forward to incorporate this model for a Multiview anomaly detection problem, where we have more than one view of the same location using different cameras. The area covered by the cameras may have overlaps or may be all the cameras are focused in

a same area but from different angles. All these Multiview approaches give more cues for detecting any suspicious activity. Hence, detecting anomaly in this Multiview model could be a significant contribution to the computer vision society, especially to the surveillance departments. The future work is mainly focused toward these directions.

APPENDIX A

MATLAB Source Codes

This Section deals with the necessary source codes developed on a MATLAB environment. The codes are easy to use and have been organized in a tabular form as follows.

Table 5: Saliency Map Generation

Source Code1: SUPERPIXEL LEVEL SALIENCY MAPS

```
% Parameter initialization for guided filtering
r = 9;
eps = 0.1^2;
% ****************Read first video frame****************
p=pwd;
cd('...\imagepath');
img1=im2double(imread('001.tif'));
cd(p);
[l1, Am1, C1, d1] = slic(img1, 1700, 40, 1.5, 'mean');
img1_gray=img1;
label_img1=unique(l1);
L1_img1 = zeros(size(l1,1),1);
for i=1:length(label_img1)
    L1_img1(i)= mean(img1(find(l1==i)));
end

P_TS=zeros(size(img1,1),size(img1,2));
P_SS=zeros(size(img1,1),size(img1,2));

% Superpixel level spatial saliency for first frame
tex_meas_img1 = LBPsup(L1_img1,Am1);
label_img1_SS = unique(l1);
ss2 = zeros(length(label_img1_SS));

for i=1: length(label_img1_SS)
    L1=L1_img1(label_img1_SS(i));
    S1=[L1];
```

```
    tex_x1=tex_meas_img1(label_img1_SS(i));
    y = find(Am1(label_img1_SS(i),:)); % neighbours of i
    Neigh_term=0;
    tex_term=0;
    count=0;
   for j=1:length(y)
        L2=L1_img1(y(j));
        S2=[L2];
        tex_x2=tex_meas_img1(y(j));
        tex=bitxor(tex_x1,tex_x2);
        bits=tex;
        b = 0;
while ( bits > 0 )
            bits = bitand( bits, bits-1 );
            b = b + 1;
end
        Neigh_term= Neigh_term + pdist2(S1,S2,'minkowski',2);
        tex_term=tex_term+b;
        count=count+1;
end
    NT(i)=Neigh_term/count;
    DT(i)=tex_term/count;
end
im_new = zeros(size(img1,1),size(img1,2));
for i=1: length(label_img1_SS)
    im_new(find(l1==label_img1_SS(i))) = (NT(i)*DT(i));
end
im_new=im_new./max(max(im_new));
P_SS=im_new;
for l=2:1:200
%%%%%  IR spectrum processing START
    srcFiles = dir('ImgFolder\*.tif');
    filename = strcat(' ImgFolder\',srcFiles(l).name);
    img2=im2double(imread(strcat(filename)));

[l2,Am2,C2,d2] = slic(img2, 1700, 40, 1.5,'mean');%generate superpixels
    img2_gray=img2;
    img2_gray(find(img2_gray==0))=.0027;

    label_img2=unique(l2);
    L2 = zeros(size(l2,1),1);
for i=1:length(label_img2)
        L2(i)= mean(img2(find(l2==i)));
end
    tex_meas_img2 = LBPsup(L2,Am2);
    DFD = zeros(size(l2,1),size(l2,2));
    label_img2=unique(l2);

    D2 = struct2cell(C2);
    x = D2(1,:,:);
    L(1:size(C2,2)) = x(:,:,:);
    L2 = cell2mat(L); % luminance
    x = D2(7,:,:);
    L(1:size(C2,2)) = x(:,:,:);
```

```
    r2 = ceil(cell2mat(L)); % r
    x = D2(8,:,:);
    L(1:size(C2,2)) = x(:,:,:);
    c2_IR = ceil(cell2mat(L)); % c

    DFD = zeros(size(l2,1),size(l2,2));
    L_DFD = zeros(length(L2),1);
for i = 1 : length(L_DFD)
        L_DFD(i) = abs(L2(i) - L1_img1(l1(r2(i),c2_IR(i))));
        DFD(find(l2==i))= L_DFD(i);
end
    DFD_IR=DFD./max(max(DFD));
    DFD_IR=DFD_IR>.1;

    Sel_Sup_img2=DFD_IR.*l2;
    label_img2=unique(Sel_Sup_img2);
    [m n]=size(Am2);
    w1=zeros(m,n);
    w2=zeros(m,n);

for i=2: length(label_img2)
        y = find(Am2(label_img2(i),:)); % neighbours of i
for j=1:length(y)
       tex=bitxor(tex_meas_img2(label_img2(i)),tex_meas_img2(y(j)));
            bits=tex;
            b = 0;
while ( bits > 0 )
                bits = bitand( bits, bits-1 );
                b = b + 1;
end
            w2(label_img2(i),y(j)) = b;
end
end
    w2=w2./(max(max(w2)));
    w2=1-w2;
% % Graph bulding on I1 for selected superpixels from DFD
    Sel_Sup_img1=DFD_IR.*l1;
    label_img1=unique(Sel_Sup_img1);
    [m n]=size(Am1);
    w1=zeros(m,n);

for i=2: length(label_img1)
        y = find(Am1(label_img1(i),:)); % neighbours of i
for j=1:length(y)
          tex=bitxor(tex_meas_img1(label_img1(i)),tex_meas_img1(y(j)));
          bits=tex;
          b = 0;
while ( bits > 0 )
                bits = bitand( bits, bits-1 );
                b = b + 1;
end
            w1(label_img1(i),y(j)) =b;
end
end
```

```
    w1=w1./(max(max(w1)));
    w1=1-w1;
%*********************TEMPORAL MATCHING ***************
    im2_full = zeros(size(img2,1),size(img2,2));
    sim=[];
for i=2: length(label_img2)   % for all selected superpixels
        y = find(Am2(label_img2(i),:)); % neighbours of i
        labels = [label_img2(i) y];
        Adj1 = zeros (50,50);
        Adj2 = Adj1;
for j = 1:length(labels)
for k = (j+1):length(labels)
if(Am2(labels(j),labels(k)))
                    Adj2(j,k)=w2(labels(j),labels(k));
end
end
end
        Adj2=Adj2+Adj2';
        [rr cc] = find(l2==label_img2(i));
        r1 = max(rr);
        r2 = min(rr);
        r = ceil ((r1+r2)/2);
        c1 = max(cc);
        c2 = min(cc);
        c = ceil ((c1+c2)/2);
% collocated superpixel centroid index
        y = find(Am1(l1(r,c),:));
        labels = [l1(r,c) y];
for j = 1:length(labels)
for k = (j+1):length(labels)
if(Am1(labels(j),labels(k)))
                    Adj1(j,k)=w1(labels(j),labels(k));
end
end
end
        Adj1=Adj1+Adj1';
        sim(i) = graph_similarity(Adj1, Adj2);
        sim_I1_IR(i)=P_TS(r,c);
end
    s=sim./max(sim);
for i=2: length(label_img2)
if s(i)<.1
            im2_full(find(l2==label_img2(i))) = sim_I1_IR(i);
else
            im2_full(find(l2==label_img2(i))) = s(i);
end
end
%%% Superpixel level spatial saliency %%%
    label_img2_SS=unique(label_img2);
    ss2=zeros(length(label_img2_SS));
for i=2: length(label_img2_SS)
        L1 = L2(label_img2_SS(i));
        S1=[L1];
        tex_x1=tex_meas_img2(label_img2_SS(i));
```

```
        y = find(Am2(label_img2_SS(i),:)); % neighbours of i
        Neigh_term=0;
        tex_term=0;
        count=0;
for j=1:length(y)
            L2 = L2(y(j));
            S2=[L2];
            tex_x2=tex_meas_img2(y(j));
            tex=bitxor(tex_x1,tex_x2);
            bits=tex;
            b = 0;
while ( bits > 0 )
                bits = bitand( bits, bits-1 );
                b = b + 1;
end
            Neigh_term= Neigh_term + pdist2(S1,S2,'minkowski',2);
            tex_term=tex_term+b;
            count=count+1;
end
        NT(i)=Neigh_term/count;
        DT(i)=tex_term/count;
end
    im_new = zeros(size(img2,1),size(img2,2));
for i=2: length(label_img2_SS)
        P_SS(find(l2==label_img2_SS(i))) = (NT(i)*DT(i));
end
    P_SS=P_SS./max(max(P_SS));
    im_new=P_SS;
    im2_N = im2_full./max(max(im2_full));

    p=pwd;
    cd('DestinationFolder\SS Maps\');
    imwrite(im_new, strcat(num2str(l),'.jpg')); % write file
    cd(p);

    p=pwd;
    cd('DestinationFolder\TS Maps\');
    imwrite(im2_N, strcat(num2str(l),'.jpg')); % write file
    cd(p);

    p=pwd;
    cd('DestinationFolder\STS Maps\');
    imwrite(im2_N+im_new, strcat(num2str(l),'.jpg'));% write file
    cd(p);

    img1= img2;
    L1_img1 = L2;
    l1=l2;
    Am1=Am2;
    img1_gray=img2_gray;
    tex_meas_img1=tex_meas_img2;
        clear L2_IRr2_IRc2_IRL;
end
```

Table 6: Weights and Features from Optical Flow & Saliency

Source Code2: OPTICAL FLOW & SALIENCY

```
k_NN = 5;
indx = 5; % The frame from which the observer starts
ap =1; bp =1; mp=3; np=1; Z=1000;% particle equation parameters
zz = [];
for i = 1:(size(targts,2)-1)
    display(['Loop entered no: ', num2str(i)]);
    im1 = imread(targts(i).imageFilename);
    im2 = imread(targts(i+1).imageFilename);
%Calculating the Optical Flow of im1 and im2.
uv = estimate_flow_interface(im1, im2, 'hs-brightness');
u = uv(:,:,1);  % Extracting the horizontal flow from uv;
v = uv(:,:,2); % Extracting the vertical flow from uv;
len=size(targts(i).objectBoundingBoxes,1);
    u_store=cell(len,1);
    v_store=cell(len,1);
for j=1:len
        parm = targts(i).objectBoundingBoxes(j,:);
        I=u(parm(2):parm(2)+parm(4)-1,parm(1):parm(1)+parm(3)-1);
            % Cropping & Extracting the bounding box region from u.
        u_store{j,1}=I; % Storing the bounded region in u_store cell
        clear I;
        I=v(parm(2):parm(2)+parm(4)-1,parm(1):parm(1)+parm(3)-1);
            % Cropping & Extracting the bounding box region from v.
        v_store{j,1}=I; % Storing the bounded region in v_store cell
        clear I;
        uv_bbox = [u_store v_store];
        mag{j,1} = sqrt(uv_bbox{j,1}.^2 + uv_bbox{j,2}.^2);
        theta = atan2d(uv_bbox{j,2},uv_bbox{j,1});
        angle{j,1} = mod(theta+360,360);
        hist_angle_and_mag{j,1} = angle{j,1};
hist_angle_and_mag{j,2} = mag{j,1};
end
    histogram_angle_and_mag{i,:} = hist_angle_and_mag;
    hist_all{i,:} = Histo(hist_angle_and_mag,20);
    clear hist_angle_and_mag magangle;
    display(['Histogram updated no.: ' num2str(i)])
hist_angle_and_mag{j,2} = mag{j,1};
end
histogram_angle_and_mag{i,:} = hist_angle_and_mag;
    hist_all{i,:} = Histo(hist_angle_and_mag,20);
    clear hist_angle_and_mag magangle;
    display(['Histogram updated no.: ' num2str(i)])
if i >= indx % index after the normal frames
        k=k+1;
        srcFiles = dir('Folder of Saliency Maps\*.jpg');
        filename = strcat('Folder of Saliency Maps\',srcFiles(i).name);
        im2_N=im2double(imread(strcat(filename)));
        len=size(targts(i).objectBoundingBoxes,1);
```

```
              crop_store=cell(len,1);
for  f=1:len
          parm = targts(i).objectBoundingBoxes(f,:);
          I=im2_N(parm(2):parm(2)+parm(4)-1,parm(1):parm(1)+parm(3)-1);
          crop_store{f,1}=I;
          clear I;
end
 crop_store_cell{k,1}=i;
 crop_store_cell{k,2}=crop_store;
 clear crop_store;
 indv_array = [1 : size(targts(i).objectBoundingBoxes,1)];
 A_sort=nrst_nghbr(targts(i).objectBoundingBoxes,indv_array,k_NN);
 sort_cell{k,1} = i;
 sort_cell{k,2} = A_sort;
%******SHOF &  SALIENCY*******%
            aa = hist_all{i,:};
            aa = [zeros(size(aa,1),ceil(Opt_zeta/2))  aa

      zeros(size(aa,1),ceil(Opt_zeta/2))];
        [~,I] = max(aa,[],2);
        SHOF=[];
for z=1:size(aa,1)
            SHOF = [SHOF; aa(z,(I(z,1)-
                  floor(Opt_zeta/2)):(I(z,1)+floor(Opt_zeta/2)))];
end
        hist_aftr_indx{k,1} = i;
        hist_aftr_indx{k,2} = SHOF;
        SHOF_all = cell(size(sort_cell{k,2},1),2);
        sal_all = cell(size(sort_cell{k,2},1),2);
for l=1:size(sort_cell{k,2},1)
            SHOF_all{l,1} = hist_aftr_indx{k,2}(indv_array(1,l),:);
            SHOF_all{l,2} = hist_aftr_indx{k,2}(sort_cell{k,2}(l,:),:);
            sal_all{l,1} = crop_store_cell{k,2}(indv_array(1,l),:);
            sal_all{l,2} = crop_store_cell{k,2}(sort_cell{k,2}(l,:),:);
            sal_x = sal_all{l,1}{1,1}(:);
            x =  SHOF_all{l,1};
for chi=1:k_NN
                y = SHOF_all{l,2}(chi,:);
                SHOF_all{l,3}(1,chi) = sum((x-y).^2 ./ (x+y+eps)) / 2;
                f(:,chi) = ((mp*ap)/((Z*SHOF_all{l,3}(1,chi))^(mp+1)))-
                           ((np*bp)/((Z*SHOF_all{l,3}(1,chi))^(np+1)));
                sal_y = sal_all{l,2}{chi,1}(:);
                H = Hausdoff_dist(sal_x,sal_y);
                sal_all{l,3}(1,chi)=H;
end
            SHOF_all{l,4}=abs(1./f(1,:));
            clear f;
for n = 1:k_NN
%% Calculating the linking weights of each targets
W_matrix(:,n) = (SHOF_all{l,4}(1,n))/sum(SHOF_all{l,4});
%% normalising saliency wts using max
norm_sal_wei1(:,n) = (sal_all{l,3}(1,n))/max(sal_all{l,3});
%% normalising saliency wts using sum
norm_sal_wei2(:,n) = (sal_all{l,3}(1,n))/sum(sal_all{l,3});
```

```
end
            SHOF_all{l,5} = W_matrix(1,:);
            clear W_matrix;
            sal_all{l,4} = norm_sal_wei1(1,:);
            clear norm_sal_wei1;
            sal_all{l,5} = norm_sal_wei2(1,:);
            clear norm_sal_wei2;
for q = 1:k_NN
       mag_nghbr = histogram_angle_and_mag{i,1}(sort_cell{k,2}(l,:),2);
              F_matrix(1,q) = max(mag_nghbr{q,1}(:));
              F_matrix(2,q) = min(mag_nghbr{q,1}(:));
              F_matrix(3,q) = mean(mag_nghbr{q,1}(:));
              F_matrix(4,q) = var(mag_nghbr{q,1}(:),1);

       mag_sal = crop_store_cell{k,2}(sort_cell{k,2}(l,:),1);
              G_matrix(1,q)=max(mag_sal{q,1}(:));
              G_matrix(2,q)=min(mag_sal{q,1}(:));
              G_matrix(3,q)=mean(mag_sal{q,1}(:));
              doub_mag = double((mag_sal{q,1}(:)));
              G_matrix(4,q)=var(doub_mag(:),1);
end
            SHOF_all{l,6} = F_matrix;
            clear F_matrix;
            sal_all{l,6} = G_matrix;
            clear G_matrix;
end
        SHOF_frame{k,1} = i;
        SHOF_frame{k,2} = SHOF_all;
        sal_frame{k,1}=i;
        sal_frame{k,2} = sal_all;
        %% Zeta Learning %%
else
        k=0;
        kk=0;
for zeta_rnge = 0:0.1:1
            zeta = round(zeta_rnge*size(hist_all{i,1},2));
            zeta = 2*round((zeta+1)/2)-1;
            kk=kk+1;
            zeta_cell{i,1} = i;
            zeta_cell{i,2} = 1:size(targts(i).objectBoundingBoxes,1);
B_sort = nrst_nghbr(targts(i).objectBoundingBoxes,zeta_cell{i,2},k_NN);
            zeta_cell{i,3} = B_sort;
            bb = hist_all{i,:};
            bb = [zeros(size(bb,1),ceil(zeta/2)) bb
                                    zeros(size(bb,1),ceil(zeta/2))];
            [~,I] = max(bb,[],2);
for p = 1 : size(zeta_cell{i,3},1);
      I1 = I(zeta_cell{i,2}(1,p),1);
      for p1 =1:k_NN
I2 = I(zeta_cell{i,3}(p,p1),1);
hist_I1 = bb(zeta_cell{i,2}(1,p),I1-floor(zeta/2):I1+floor(zeta/2) );
hist_I2 = bb(zeta_cell{i,3}(p,p1), I2-floor(zeta/2):I2+floor(zeta/2) );
zeta_cell{i,4}(p,p1) = sum( (hist_I1-hist_I2).^2 ./
```

```
                                        (hist_I1+hist_I2+eps)) / 2;
zeta_cell{i,5}(p,1) = var(zeta_cell{i,4}(p,:),1);
zeta_cell{i,6}(:,kk) = [zeta; min(zeta_cell{i,5})];
end
end
 zeta_all{i,kk} = {zeta_cell{i,1:5} zeta_cell{i,6}(:,kk)};
end
        zz = cat(2, zz, zeta_cell{end,6});
        [Min,zI] = min(zz(2,:));
        Opt_zeta = zz(1,zI);
end
end
```

Table 7: Fusion of the two Motion cues

Source Code3: FUSION OF THE TWO MOTION CUES

```
fused_wt = {};
frame = indx;
for i = 1:size(SHOF_frame,1)  %% for the number of frames of SHOF
for j = 1:size(SHOF_frame{i,2},1)  %%for the number of individuals in
each frame
     W1 = SHOF_frame{i,2}{j,5}; % Motion Cue of an individual from OF
     W2 = sal_frame{i,2}{j,3}; % Motion Cue an individual from
Saliency
        X_bar = (W1+W2)/2; % A vector bisecting both wt. vectors
        x = X_bar/sqrt(X_bar*X_bar'); % Unit vector of X_bar
        fused_wt{j,1} = 2*(X_bar*X_bar')*x;

for n = 1:k_NN
for f = 1:size(SHOF_frame{i,2}{j,6},1)
                W3 = sal_frame{i,2}{j,6}(f,n);
                W4 = SHOF_frame{i,2}{j,6}(f,n);
                Y_bar = (W3+W4)/2;
                y = Y_bar/sqrt(Y_bar*Y_bar');
                f_ftr(f,n) = 2*(Y_bar*Y_bar')*y;
end
end
        fused_ftr{j,1} = f_ftr;
end

    Fused_frame{i,1} = frame; % Storing frame no.
    Fused_frame{i,2} = fused_wt; % Storing Fused Wt.
    Fused_frame{i,3} = fused_ftr; % Storing Fused Features.

    clear fused_wtfused_ftrf_ftr;
    frame = frame+1;
end
```

Table 8: Anomaly Detection

SourceCode4: ANOMALOUS FRAME DETECTION

```
for i = 1:size(beta_cell,1)-2 % for frames marked aftr #indx, i.e. aftr
                                  the normal frames
    fval = [];
for j=1:length(beta_cell{i,2})
if isempty(find(beta_cell{i+1,2}==beta_cell{i,2}(1,j), 1)) == 0
      % observer index corresponding in nxt frame
        indx = find(beta_cell{i+1,2}==beta_cell{i,2}(1,j));
            f1 = Fused_frame{i,3}{beta_cell{i,2}(1,j),1};
            f2 = Fused_frame{i+1,3}{indx,1};
            w1 = Fused_frame{i,2}{beta_cell{i,2}(1,j),1};
            w2 = Fused_frame{i+1,2}{indx,1};
           [x fval] = emd(f1', f2', w1', w2', @gdf);
end
end
    d_val{i,1} = beta_cell{i+1,1}; % frame no of i+1 frame
    d_val{i,2} = 1-fval;
    d_val{i,3} = 1-(sum(d_val{i,2})/length(fval)); % AD^(t+1)
end

a =  [d_val{:,3}]>=0.5;%anomalous frame nos. having d_val>thhreshold
an_fr = [];
for i = 1:size(beta_cell,1)-2

      if a(1,i)==1
        an_fr = [an_fr;d_val{i,1}];
      end

end
```