University of Alberta

A Novel Framework for Unique People Count from Monocular Videos

by

Satarupa Mukherjee

A thesis submitted to the Faculty of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

Department of Computing Science

Spring 2014
Edmonton, Alberta

Abstract

Counting unique number of people in a video (i.e., counting a person only once while the person passes through the field of view (FOV)), is required in many video analytic applications, such as transit passenger and pedestrian volume count in railway stations, malls and road intersections, aid in security and resource management, urban planning, advertising and many others.

In this PhD thesis I have developed a robust algorithm to generate unique people count from monocular videos taken from an arbitrary angle. From applications point of view, my algorithm is one of the most economical ones, because it can work with existing video cameras already mounted. Within a region of interest (ROI) on the FOV of the camera, I compute influx/outflux rate of people, i.e., number of people coming in or going out of the ROI per unit time. Then, I sum the influx/outflux rate between any two time points to estimate the number of people that entered and/or left the ROI within that time interval. I employ two well-known computer vision techniques for this purpose: Gaussian process regression (GPR) to estimate the number of people present within a ROI and optical flow-based tracking of the boundary of the ROI.

The principle roadblock in most of computer vision problems is occlusion. To avoid this bottleneck, we adopt the combination of (a) the concept of influx and outflux of fluid mass from computational fluidics, (b) the GPR to estimate the number of people within a ROI and (c) ROI boundary tracking (as opposed to object or feature tracking) for a short period. Thus, the principal contribution of the thesis is to successfully handle occlusions by computing the average influx and/or outflux of people and avoiding people detection and tracking.

We validate the proposed algorithm on 19 publicly available monocular bench-

mark videos. Occlusions are abundant in these videos, yet we obtain more than 95% accuracy for most of these videos. We also extend our proposed framework beyond monocular videos and apply it on multiple views of a publicly available dataset with about 99% accuracy.

Keywords: people counting, occlusion, boundary tracking.

Acknowledgements

I would like to extend my sincere appreciation to my supervisor Dr. Nilanjan Ray, for his continuous help, encouragement and guidance without which completion of my thesis would remain a far cry. I also thank him for giving me the opportunity to work on on the exciting field of people counting. I would always be grateful to him for his patience and encouragement whenever I sought advice. I believe that he has had a everlasting influence on my career.

Next, I would like to thank my co-supervisor Dr. Hong Zhang who always ushered me with his valuable input, and had a great influence throughout the course of my PhD. I always hailed him as the biggest critic of my work and appreciated his straightforwardness during constructive criticism of my work. Special thanks to my supervisor committee Dr. Pierre Boulanger and Dr. Dale Schuurmans for critical review of my work and providing valuable input to steer my work towards completion.

I should also mention thanks to my past and present colleagues at the CIMS (Centre for Intelligent Mining System) lab for providing me friendship, inspiration, advice and encouragement throughout my PhD. Furthermore, I extend my gratitude to Department of Computing Sciences for providing me with state-of-the-art facilities and MITACs Accelerate for giving me the scholarship opportunity and a rare chance of working with the city of Edmonton.

I should also mention some names whose continuous support led me through the rigorous process of my PhD: viz my parents and grandparents, my friends Anupam Haldar and Li He and my dearest neighbours and guardians Dr Reinheild Boehm (retired associate professor, Faculty of Extension) and Dr Judith Golec (retired professor and Dean of Sociology, Department of University of Alberta).

Finally, I would like to thank Dr. Yang Cong for providing me with the LHI dataset. I also acknowledge the following sources of funding for my work: NSERC, AQL Management Consulting Inc., and Computing Science, University of Alberta.

Table of Contents

List of Tables

List of Figures

List of Symbols

UIOC	Unique Influx Outflux Count
DTV	Detection Tracking and Validation
LRT	Light Railway Transit
MH	Motion Histogram
STG	Spatio-Temporal Gradient
STV	Spatio-Temporal Validation
ROI	Region of Interest
ROC	Receiver Operating Curve
FOV	Field of View
SVM	Support Vector Machine
SVR	Support Vector Regressor
GP	Gaussian Process
AM	Approximate Median
MG	Mixture of Gaussians
ViBe	Visual Background Extractor

HS	Horn Schunck
LK	Lucas Kanade
BA	Black Anandan
OF	Optical Flow
TP	True Positive
FP	False Positive
FPR	False Positive Rate
TPR	True Positive Rate
HOG	Histogram of Oriented Gradients
LBP	Local Binary Patterns

Chapter 1

Introduction

1.1 Problem Statement and Motivation

People counting is important for solving many important applications like traffic management, detection of overcrowded situations in public buildings, tourist flow estimation, surveillance and many other scenarios. It is also a significant component in video analytics. By **unique people count**, we mean the computation of the total number of people in a specific time interval by counting a person only once while the person is present within a field of view (FOV) or a region of interest (ROI) within the FOV.

Among the different real life applications of people counting, one of the most important is the traffic management system. An automatic people count system would help transit authorities to optimize frequencies of transit vehicles at different times of the day by taking into account the total people count during those times. In this way, emission of greenhouse gases can be prevented which leads to the welfare of mankind and the environment. Traffic and transit management also helps in different resource management and urban planning in broader aspects. Cumulative people count is effectively used in advertising purposes and store management. Store managers may need to monitor people count for different reasons. One of them may be to learn how frequently the store is visited and at what times. This type of information helps the store manager to schedule employees efficiently. Moreover, a people counting software may also be able to reciprocate the rate of increase of number of customers due to the new advertisement campaign. With

the aid of a real time people counting software, shop assistants can be assigned to necessary areas of a store in an interactive way. For public transportation, an automatic people counting system may be again needed for determining the optimal schedule. Overcrowding situations can also be determined with such a software by estimating the number of people in large crowds. Thus this software can be used in calculating the total number of people participating in a demonstration or festival or even tourist flow estimation during certain seasons of the year. Finally, a very important utility of a people counting software lies in the domain of surveillance videos. Most of surveillance applications need the total count of people during certain interval of time to ensure safety, security, support site management and many others. Some crucial instances include estimation of queue length in retail outlets, monitoring entry points in secured buildings, train stations, bus terminals and even in military camps. Thus counting unique number of people plays a very important role in modern technologies.

People counting systems can be roughly categorized into computer vision based and non-computer vision based techniques. The non-computer vision based systems use many different technologies [5], each with its own advantages and disadvantages. Probably the most straightforward system is the tally counter or clicker counter. It has a very simple working mechanism where pressing a button activates the count. However, the method needs human intervention, which is both labour and cost intensive. A very accurate people counting system is the mechanical counter, known as the turnstile, which needs to be turned by the individual each time he/she crosses it in order to take into account the individual count. However again, this method is invasive and disruptive. Laser beam-based sensors are among the non-invasive methods used frequently in railway stations. These methods are inexpensive, but they are not suitable for counting people in outdoor environments, because their performance can be negatively affected when subjected to direct sunlight. Another well-known non-invasive people counter is based on thermal sensors. However, once again, they are sensitive to ambient temperatures.

Computer vision-based solutions to date are mainly based on methods that use either a camera network or a monocular video. The network of multiple cameras

is one of the most advanced technologies used for people counting. It takes into account different views of people with different camera angles to avoid occlusion. But the setting up of the system can be costly and the process may often be cumbersome due to lack of resources. Moreover, homography constraints often need to be applied [4] for finding out correspondences among views of people obtained from multiple cameras in order to perform any kind of tracking or counting. The homography computation may also lead to the occurrence of transfer errors (summation of the projection error in each camera view for a pair of correspondence points) that needs to be dealt with. Our proposed approach to finding the unique people count is based on monocular videos. Our principal motivation is to make use of existing cameras and avoid expensive camera network setup and maintenance.

The computer vision based algorithms for people counting from monocular videos are mainly used for finding out two types of counts - frame based people count and unique people count.

The frame based people counting algorithms count people in individual video frames with reasonable accuracy even in the presence of occlusions [8, 9, 18, 59, 67, 17]. These methods use extracted features from individual frames and count the number of people in each frame with the help of machine learning techniques that map the extracted features to the number of people present in the frame. But these methods fail to count the unique number of people present in a video over an interval of time, as they do not consider the correspondence of the same person over multiple frames. For example, if there are n people in the first frame and one person enters, while another person exits the FOV in the second frame, the frame based counting will produce n as the people count for the second frame. However, the unique count of people for the two frames should be $n + 1$.

The computer vision based solutions to unique people count can be further categorized into three types:

(a) the detection and tracking based approach [28, 34, 64]

(b) the visual feature clustering based approach [6, 57]

(c) the line of interest (LOI) count approach [40, 16, 35]

The detection and tracking based approaches [28, 34, 50, 64] count people by detecting individuals on an image and creating corresponding trajectories by tracking them. The number of trajectories in an interval of time accounts for the number of people. The method is efficient when the object size is large, with sparse crowd and limited or no occlusion, because large object size helps in the detection due to the presence of enough image pixels depicting the object. Tracking is successful for overhead FOVs where little or no occlusion is present, but succumbs to failure in case of whole body views, where partial occlusion is present. Applying the detection-tracking approach becomes difficult in dense crowds, where each person is depicted by only a few image pixels and people occlude each other.

The visual feature trajectory clustering methods [6, 57] cluster feature trajectories that exhibit coherent motion and the number of clusters is used as the number of moving objects. This type of method requires sophisticated trajectory management, such as, handling broken feature tracks due to occlusions or measuring similarities between trajectories of different lengths. Thus, in crowded environments, it is frequently the case that coherently moving features do not belong to the same person. Thus, equating the number of people to the number of trajectory clusters becomes quite error prone. Once again, occlusion is a serious bottleneck for these methods too. Thus, the first two individual based analyses are somewhat successful for low density crowds or overhead camera views, but they are not competent enough for large crowds.

In the LOI counting methods [40, 16, 35] the basic principle is to create a line of interest within the FOV and take into account people crossing the LOI. At first, temporal images are constructed across the LOI over a period of time. Features are extracted from the temporal images and the cumulative people count is estimated by using a regression function. The disadvantage of these methods is that the feature computation pipeline used in these methods is fairly complicated. These methods may not perform well if the walking speed varies greatly within the crowd. Finally, a suitable LOI need to be drawn within the FOV to take into account all the people within the FOV. Drawing this type of LOI is problematic for FOVs where people are moving in multiple directions.

In this PhD thesis I have developed a robust algorithm to generate the unique people count for monocular videos taken from an arbitrary angle. From an applications point of view, my algorithm is one of the most economical ones, because it can work with existing video cameras already mounted. To avoid the expensive and also challenging video camera network system, my algorithm should work on the view taken from a single camera. Finally, apart from dealing with sparse crowds, the algorithm is able to deal with large as well as dense crowds. Hence, it should be capable of handling occlusions.

1.2 Proposed Solution

As discussed in the previous section, unique people counting finds its use in a variety of applications. But the methods used are either non-computer vision based techniques (both mechanical and electronic) which are either too cumbersome or costly, or are restricted by the environment in their applications. The computer vision based techniques currently in work or in research are mostly affected by occlusion or need a network of cameras. A network of cameras capture different views of people from different angles. This often helps to deal with occlusion problem which is a significant hurdle in the domain of computer vision. Moreover, homography constraints are also needed while we work with multiple camera views. But this method utilises costly and complicated setup, which needs skilled staff to maintain and manage, that restricts the use of camera network to only big enterprises or corporations. This leaves the small businesses unable to use computer vision based counting methods for their operations. Thus, an affordable solution is necessary to count people from monocular videos plagued with occlusion, which could be adapted for use in existing camera setup.

So, I aim to build a unique people counting software which can use any existing camera setup, i.e., it should mainly work on monocular videos. Expensive solutions to people count problems prohibit many end users of small scale businesses to use them. So, my purpose is to build an inexpensive solution so that it is adaptable to any type of applications and still provide a good solution. In order to avoid

expensive solution, we exclude multiple views of humans captured from camera network from our input data. Instead we work with monocular videos so that our solution can be provided with any existing cheap single camera setup. We also keep in mind, that the solution should provide satisfactory accuracy in people count on any camera angle without using multiple human views and still should handle occlusion. As the solution can be provided using a single camera, the setup is simple and maintenance cost is low which is aptly suitable for small business and several real life applications.

I initially worked with the City of Edmonton for developing an automatic software for counting passengers in the LRT stations of Edmonton. The local transportation agency in Edmonton required automatic software capable of counting the number of LRT passengers. Such software can serve their future planning and decision making system. Transit count data is an important component in Transit and Transportation Planning as they are an input for the design of effective and efficient transit service and transportation management systems. The count data helps to analyse the impacts of changes to the transit service, thus optimizing the frequency of transit vehicles which helps to save fuel and human resources. This data also decreases emission of green house gases from the vehicles impacting the environment. Moreover, the different people count in different times of the day will help the city in further decision making and design of transportation system in future. Recently, most of the data collection processes of the City of Edmonton use either a manual counting system or laser beams for the LRT. Manual counting is expensive and intermittent, whereas laser beams are cheap but not suitable for outdoor environments. So, the city expressed zeal to develop a new vision for a Transit Monitoring application.

In order to achieve this goal, the City of Edmonton used existing cameras with monocular videos to lower the expense. The different views captured can be broadly classified into top views and full body views. In my research work, I initially produced a framework for counting passengers in LRT stations, that has three major steps: people detection, tracking and validation. The framework is tested on both top views and whole body views of passengers. In the top views of passengers,

the Hough circle based detection algorithm [27] is used to detect people, then an optical flow based tracking algorithm [29] is evoked to track each person detected in a frame, and finally all the trajectories resulted from the tracking algorithm, are sieved through a spatio-temporal validation algorithm to verify whether the trajectory followed a person correctly [50]. The total number of valid trajectories gives the total number of unique count of the people within a specific time interval.

Apart from top views, the second focus was to work with whole body views of people. In this case, a background subtraction method is proposed for the initial detection of a human being, an optical flow method for tracking and a motion histogram based technique for classifying the trajectories into human or nonhuman. Although different methods are proposed for the framework, the framework is not constrained by these methods. For example, in the case of a sparse crowd, background subtraction may be used for the detection process, while for dense crowds, a histogram of oriented gradients can be applied. Thus this framework is flexible enough for future work.

The speciality of this framework is the introduction of the validation step where the trajectories are classified into human or non human after the detection and tracking work are completed. Most of the existing algorithms skip the validation step. Here we argue that the validation step is crucial in this type of people counting work as most of the existing detection algorithms produce a significant number of false alarms which may lead to a wrong count.

The task of people counting with the above mentioned detection-tracking-validation (DTV) algorithm becomes complicated when the crowd is very dense and there is significant occlusion. Occlusion handling is a challenge in the domain of human recognition and tracking problems when each human in a crowd is analysed individually. So, for occlusion handling, the whole crowd needs to be analysed instead of individual analysis.

In my PhD thesis, the novel framework I have developed monitors a human crowd globally, avoiding individual analysis. The input to this framework, is a monocular video consisting of human views and the output is the total unique count of people within a certain duration of the video. The aim of the framework is

an application towards real life problems. It is termed novel because it does not resemble any of the above mentioned three methods for unique people count. First of all, the method analyses the crowd globally unlike individual analysis performed in the detection-tracking or visual feature trajectory clustering methods. This helps to avoid the occlusion which plagues the individual detection and tracking work. Secondly, the method differs from the LOI counting methods, because it does not rely on any temporal image generation and their analysis. It works with ROIs which can be of any shape, rather than only a straight line as in the LOI methods. Thirdly, we incorporate concepts, such as influx, outflux and boundary tracking in the field of people counting, which when combined with a non linear regressor help to handle occlusion better than the LOI counting and the detection-tracking methods.

To achieve unique people count within a ROI (rectangular or any other shape) on the FOV of the camera, we compute influx/outflux rate of people, i.e., number of people coming in or going out of the ROI per unit time. Then, we sum the influx/outflux rate between any two time points to estimate the number of people that entered and/or left the ROI within that time interval. Thus, we are able to compute the influx and/or the outflux rate of unique people at any time instant. Summing these rates between any two time points provide us with the unique people count. In addition to unique people count, we are also able to count the number of people moving in different directions by taking into account directional influx/outflux along the edges of the ROI. To compute influx and outflux, we use machine learning to estimate the number of people within a region of interest and we track the boundary (as opposed to tracking any object, blob, or feature) for a *short time period*. In addition, any remaining effect of occlusions is mitigated by averaging the influx and outflux rate over a period of time. Our framework is online and it does not accumulate error over time. The reported running times make it suitable for realtime applications. We have termed this framework as **Unique Influx Outflux Count (UIOC)** framework.

For validation of the UIOC framework, we have performed extensive experiments on four benchmark datasets: the UCSD dataset [8], which consists of a one hour video of 25,656 frames, the FUDAN dataset [59] which has 1500 frames, the

LHI dataset [16], which has 12 videos captured at different camera angles ($90^o, 65^o$ and 40^o), and of durations ranging between 5 to 15 minutes and the PETS 2009 dataset [76] which consists of multiple camera views targeted at the evaluation of various surveillance applications. Although the focus of the current work is the application of people count, we have also tested our algorithm for counting cells from video microscopy [54].

The high accuracy in performance achieved on all of the videos, accompanied with the attributes of fastness and capability of working on multiple views with competence, endows our framework with applicability to any type of video and make it suitable for various commercial applications.

1.3 Contribution of Thesis

In this thesis work I have introduced a novel framework named **UIOC** as described in the previous section. The novel framework designed here for unique people count, is able to count people with excellent accuracy overcoming occlusion. This framework contributes to the scientific community in the following ways:

(a) Better occlusion handling - Occlusion is one of the major problems in the domain of computer vision. The effect of occlusion is handled competently in the proposed framework due to three factors -

  (i) Boundary tracker, which computes pixel motion only on the ROI boundary, thus works with a very small set of pixels and also for a short period of time.

  (ii) Machine learner-based frame count can handle occlusion to a great extent.

  (iii) Remaining effects of occlusion overlooked by the machine learner are resolved by averaging the influx and outflux rate over a period of time.

  The novelty here is that, the framework does not resemble any of the existing methods in literature ie detection-tracking framework, visual trajectory

clustering framework or LOI counting framework. It avoids individual detection and tracking but is still capable of overcoming occlusion with dexterity. Though it uses a machine learner used in existing literature for occlusion handling, its better occlusion handling capacity is not only because of the machine learner but also due to incorporation of the boundary tracker and averaging of influx and outflux rate over a period of time which empowers it with better occlusion handling capacity.

(b) Versatility in application - Our framework is capable of working on any existing camera setup as it can work on both monocular and multiple camera views. It is also flexible to use on views captured at different angles overcoming any amount of occlusions. The results presented in the following chapters of the thesis support our claim. As the system is able to work on monocular videos, it is aptly suitable for small businesses and various real life applications. In addition to people counting, the proposed system is successfully applied to cell counting with remarkable accuracy [54] which depicts its versatile nature.

(c) High Accuracy - The system produces high accuracy on 19 publicly available benchmark videos. Apart from that, it also produces more that 90% accuracy on the LRT dataset provided by the City of Edmonton which illustrates its utility in real life problem solving.

(d) Speed - The framework is very fast and works as fast as 30 frames per second on a system with Intel(R), core(TM), DuO CPU, E8400 @ 3GHz which is real time. The system is implemented in OpenCV using the MATLAB implementation of the GP.

(e) Online - The system is totally online in nature and is capable of working on streaming videos.

(f) Non-accumulation of errors - A remarkable characteristic of our system is that, it does not accumulate any error with time. Thus it can be applied on

any length of video sequences with satisfactory performance. This fact is validated through our results.

(g) Directionality - Directionality is incorporated very easily in our framework. This enables us to count the number of people moving in any direction within the field of view. This attribute is extremely helpful in real life applications like traffic management.

1.4 Organization of Thesis

The thesis is organized as follows. First, the background and existing work on people counting have been explored in Section 2. In Section 3, we provide a detailed description of the two types of frameworks which we propose for the people counting task along with the results obtained. Section 4 illustrates the flexibility of the UIOC framework with the help of a discussion of its application on cell counting. We conclude in Section 5 with probable future works.

Chapter 2

Background and Related Work

The computer vision based algorithms for people counting from monocular videos are mainly used for finding out two types of counts: the frame based people count and the unique people count.

2.1 Frame Based People Count

The frame based people counting algorithms count people in individual video frames with reasonable accuracy even in the presence of occlusions [8, 9, 18, 59, 67, 17]. These methods use extracted features from individual frames and count the number of people in each frame with the help of machine learning techniques that map the extracted features to the number of people present in the frame. These approaches for people counting are currently very popular in computer vision as they monitor the entire crowd environment globally without individually detecting or tracking the humans. Hence, the problem of occlusion is handled very deliberately with the help of machine learners instead of tackling it in the detection or tracking phase. These methods are supervised where in the training phase, low level features are extracted from each training image. Then a machine learner is trained to learn a relationship between the extracted features and people count in each frame. In the testing phase, features are extracted from each test frame and the machine learner obtains the count in each test frame based on the extracted features.

In [8], the authors initially use mixture of dynamic textures [11] to segment regions having motion. Then in the training phase, they extract features from each

frame and calculate people in each frame. With the features and people count from the training images, they train a Gaussian process regressor [58] to obtain a relationship between the extracted features and people count. Then in the testing phase, they extract features from test images. Finally, they estimate the people count on each image with the learned function of the trained regressor using the extracted features.

In the paper, [9], the authors have extended the method discussed in [8]. The Gaussian process regressor, which they used previously, produces real valued outputs, whereas the actual counts are discrete. So, in [9], the authors overcome this limitation by using a Bayesian treatment of Poisson regression [10].

In [18], the authors use SURF points as features. They extract SURF points from each frame of the input video. For each point on which a surf feature is detected, they estimate motion vector w.r.t the previous frame using a block matching technique. Then points with null motion vectors are pruned. The remaining points are divided into clusters using a graph-based clustering method as they worked on videos where people are walking in groups. A support vector machine is trained with the number of points in a cluster and the distance of the cluster from the camera to estimate the number of people in the cluster.

In the paper [17], suitable scale-invariant features are extracted from image frames and the moving feature points are chosen taking into account that they correspond to moving people. The frames are divided into a number of horizontal zones to avoid perspective distortions. The people count in each frame is counted by summing up the people counts in all the horizontal zones.

The method used by Tan et al. [59] is almost similar to Chan's method [8]. Unlike Chan's method, here the authors used semisupervised learning to reduce the learning time. They add the unlabeled data as regularization term to refine the performance of learning. Elastic net [69], which is a variant of Lasso [60], is used as a machine learner to learn the mapping function between the extracted features and the people count. The authors chose Elastic net as their machine learner because of its capability of reducing some redundant features.

Zisserman *et al.* proposed a novel method [67], which estimates image density

and integrates this image density over any image region to produce object count within that region. The authors have used this method for people count in individual image frames along with other object count applications.

2.2 Unique People Count

The computer vision based solutions to unique people count can be further categorized into three types: a) the detection and tracking based approach [28, 34, 64], b) the visual feature clustering based approach [6, 57], and c) the line of interest (LOI) counting approach [40, 16, 35]. The first two individual based analyses are somewhat successful for low density crowds or overhead camera views, but they are not competent enough for large crowds. In these types of views, there is too much occlusion, or people are depicted by only a few pixels, or the situations are too challenging for tracking. The LOI counting methods are capable of handling occlusion, but these methods have received relatively less attention so far.

2.2.1 Detection-Tracking Approach

The detection tracking approach counts the number of people by detecting individuals in an image and creating corresponding trajectories by tracking them. The number of trajectories in an interval of time accounts for the number of people. There are lots of human detection methods available in the literature. The most simple one is background subtraction method [34] where moving human bodies are detected as silhouettes. Background subtraction is suitable for the situations of stationary cameras where the crowd is sparse and each connected foreground object, known as a blob, corresponds to a single object. But sometimes the blobs may be fragmented for low contrast or partial occlusion by humans and also there may be some non-human pixels included in the blob due to shadows and noise. Hence, some morphological post processing is often needed. In the case of dense crowds, the segmentation task becomes even more complicated as one big blob consists of several humans. So, human segmentation cannot be performed from the blobs as they do not provide direct object level description. In order to segment this type of

foreground, where human objects overlap with each other, a Bayesian framework is needed [26, 65]. Zhao *et al.* [65] pose the problem of human segmentation as a model-based segmentation problem in which human shape models are used to interpret the foreground in a Bayesian framework. A number of 3D human models are used to capture the gross shape of standing and walking humans. The human shape is modelled by four ellipsoids corresponding to head, torso and two legs. An ellipsoid fits human body parts well and its projection is an ellipse which has a convenient form. Each ellipsoid has two parameters named length and fatness. Therefore, the parameters of each human object are model/orientation, position, height and fatness, respectively, which are used for the Bayesian segmentation. On the other hand, Ge *et al.* [26] tackles both the problems of detection and counting of people in crowded environments by a Bayesian mark point process model. The model uses a spacial stochastic process to take into account the number and placement of individuals with a conditional mark process to select body shape. Initially, they estimate a mixture of Bernoulli shape prototypes with an extrinsic shape distribution, which describes the orientation and scaling of the shapes for any given image location. Then they automatically learn a mark process from the training video. The maximum a posteriori configuration of shapes is efficiently estimated with the help of a Markov Chain Monte Carlo framework. This leads to very accurate estimate of count, location and pose of each person in the scene.

Another type of detection algorithm includes detection with a cascade of classifiers [62]. It develops a pedestrian detection system which takes into account both image intensity information and motion information. The overall detection procedure works as a degenerate decision tree, which is called a cascade. A positive result from the first classifier triggers the evaluation of a second classifier which has also been adjusted to achieve very high detection rates. A positive result from the second classifier triggers a third classifier, and so on. A negative outcome at any point leads to the immediate rejection of the sub-window. Stages in the cascade are constructed by training classifiers using AdaBoost and then adjusting the threshold to minimize false negatives.

Skin color information provides a robust information for detection purposes in

a cluttered environment, as used in [28]. Here the authors depict a counting system for transport vehicles, integrated in a video surveillance product. The detection is based on skin color information of faces. An iterative process is used to estimate the position and shape of multiple faces in images, and to track them. The trajectories are then processed to count people entering and leaving the vehicles.

A well known approach in the field of human detection in recent studies is the sliding window based approach. Examples include the popular Histogram of Oriented Gradients (HOG) method [20] and the Local Binary Patterns (LBP) method [49]. In [20], the authors study a novel feature descriptor for human detection, known as the HOG descriptor, which is based on fine scale gradients and fine orientation binning in overlapping descriptor blocks. After reviewing previous edge and gradient based descriptors and performing experiments on more challenging datasets, the authors prove the competence of their robust object detection method.

LBP [49] is another well-known sliding window approach which works on grayscale images and mainly carries texture information. It is a texture operator which labels the pixels of an image by thresholding the neighborhood of each pixel and considers the result as a binary number. The histogram of the 256 different labels can then be used as a texture descriptor. The most attractive characteristics of LBP are its invariance to monotonic gray-scale changes, low computational complexity and convenient multi-scale extension. The philosophy behind LBP is simple and elegant, unifying statistical and traditional structural methods. The advantage of LBP over HOG is that, when the target object typically appears in a cluttered environment and the unexpected noises drastically degrade the performance, only gradient information is insufficient to judge useful points and outliers. In this case, the concept of uniform LBP provides the possibility for effectively removing outliers. Moreover, LBP is more robust to illumination changes.

In a recent people counting paper [64], the authors used both the HOG and LBP descriptors for human detection purpose. The method designed a robust head-shoulder detector for people counting in surveillance systems. Initially for detection purpose, they combined the HOG and LBP as feature set. Principal Component Analysis (PCA) is used to reduce the dimension of the HOG-LBP feature set. Then

they incorporate the detector through a particle filtering tracking to get the final count. This method performed well for partial occlusion as the detection method was robust and it was based on only head and shoulders and not on the whole body.

As the above mentioned methods detect whole pedestrians or a part of the pedestrian (face or head-shoulder), they are not very effective for dense crowds where there is significant occlusion. These types of scenarios have been handled by part based detectors [63, 23]. In [63], an individual human is modelled as an assembly of natural body parts (head, shoulder, torso, legs). The authors introduce edgelet features which are a new type of silhouette oriented features. Part detectors are learned by a boosting method depending on the edgelet features. Responses of part detectors are combined to form a joint likelihood model that includes instances of multiple, possibly inter-occluded humans. The human detection problem is then formulated as maximum a posteriori (MAP) estimation. On the other hand, [23] describes an object detection method based on mixtures of multiscale deformable models which has been applied on many objects other than human beings. The novelty of the detection method is that it uses a star-structured part-based model defined by a root filter along with a set of part filters and associated deformation models. As it is also a part based detector, it is capable of handling significant occlusion in dense crowds where the detection is based on visible body parts even if some of the body parts remain occluded.

In the detection-tracking framework, after a person gets detected by a detection algorithm, the detected person needs to be tracked throughout the field of view for generating a trajectory. The number of generated trajectories accounts for the number of people in the video. The different tracking algorithms available in literature include block matching technique, optical flow methods [29, 39, 7], and different particle filter methods.

Before proceeding to the next section, a brief discussion about these tracking methods is needed. Among the various tracking methods available in the literature, the different optical flow (OF) methods have gained popularity in recent years. One of the oldest of them is the Horn-Schunck (HS) [29] method. This method generates the optical flow pattern by assuming that the brightness pattern varies smoothly

almost everywhere in the image. [The theory of optical flow can be found in Appendix]. Thus this method always attempts to minimize distortions in flow and preferably tries to produce solutions which show more smoothness. Some remarkable attributes of this OF method are that it is very fast in nature and robust against quantizations of brightness levels and additive noise. Due to its fast nature, I have used it in many stages of my work.

The Lucas Kanade (LK) [39] method solves the aperture problem of the OF problem in another way. It works on the assumption that the displacement of the image pixels between two nearby frames, is too small and almost constant within the neighbourhood of a point under consideration. Thus the method assumes the OF equation to be valid for all pixels within a window centred at that pixel. Finally, the LK method solves an over-determined system by least squares principle [38].

The Brox method [7] of optical flow models the energy functional of the OF under three assumptions : brightness constancy assumption, gradient constancy assumption and a discontinuity-preserving spatio-temporal smoothness constraint. The numerical scheme designed by this method is very consistent and provides a theoretical foundation to warping. The method is hardly sensitive to parameter variations and very robust towards noise.

One of the most accurate OF methods is the one designed by Black-Anandan (BA) [1]. This method designs a very robust and accurate solution to the OF problem by avoiding the single motion assumptions adopted by most of the state of arts and considering the effects caused by multiple motions. Thus it is applied successfully to three standard techniques of OF solutions : area based regression, correlation and regularization with motion discontinuities.

Apart from OF methods, the particle filter based trackers are also very popular due to their accuracy in performance. Among the different particle filter trackers, a recent and very fast method which needs to be mentioned here is the particle filter tracker with classifiers [12]. The speciality of this method is that, it uses classifiers for the observation function of the particle filter. Thus the likelihood function is modelled directly from the output of classifiers. The classifiers used here are the Support Vector Machine [19] and Adaboost [61]. This method is very fast and

real time in nature. Thus, it is used in many popular applications like pedestrian tracking and counting [64].

2.2.2 Visual Trajectory Clustering Approach

In the broad classification of people counting methods, the second category specifies the visual feature trajectory clustering method [57, 6]. These methods cluster feature trajectories that exhibit coherent motion and the number of clusters is used as the number of moving objects. Rabaud *et al.*'s [57] method is based on a highly parallelized version of the KLT (Tomasi-Kanade features) [3] tracker for processing the video into a set of feature trajectories. For determining the number of objects at a given time, the features present at that time are clustered into plausible objects, and the number of resulting clusters gives the count. At each time step, the present features form the nodes of a connectivity graph, whose edges indicate possible membership to a common object. Thus the problem can be modelled as an instance of graph partitioning problem with binary edge weights that can be solved using cues and various techniques described in the paper.

Brostow et al. [6] describes an unsupervised data driven Bayesian clustering algorithm which detects individual entities as its primary goal. In the paper, the authors track simple image features, eg corners and Tomasi-Kanade features and probabilistically group them into clusters. The clusters represent independent moving entities. Thus the number of clusters denotes the number of human beings.

2.2.3 LOI Counting Approach

In the LOI counting methods [40, 16, 35] the basic principle is to create a line of interest within the FOV and take into account people crossing the LOI. In [40], a LOI is drawn in the FOV first. Spatio-temporal images are formed across this LOI. Features are extracted from these spatio-temporal images. Then the number of people is counted using a regression function.

In [16], a LOI is created in a similar way. Then a ROI is created across the LOI. Pedestrians are regarded as fluid flow across the line and a novel model is designed to estimate the flow velocity field. Dynamic mosaics are constructed by integrating

over time for counting the number of pixels and edges crossing the line. Finally the number of pedestrians is estimated by applying a quadratic regression function on the dynamic mosaics.

The authors in [35] consider a virtual gate as their LOI. Low level features are extracted across the LOI using foreground pixels and motion vectors. Then the number of pedestrians are estimated using the accumulated features.

2.3 Background Subtraction Methods

Background subtraction is an integral part in different people counting algorithms. We have used background subtraction in both the frameworks proposed in this work - Detection-Tracking-Validation (DTV) [50] and Unique Influx Outflux Count (UIOC). In this section, we will discuss the different background subtraction algorithms which have been mainly considered in this work.

Among the different background subtraction algorithms, a simple and straightforward one is the Approximate Median (AM) [41] method. This method is a combined form of image differencing with respect to a median background and a Laplacian operator. The first step in this method is image differencing. Each consecutive image is subtracted from a time averaged reference image. The difference image produced as an output of this step is thresholded. This threshold is the only tunable parameter in the AM method which can be tuned with only a few training frames. Moving object pixels having values more than the threshold value are considered as foreground pixels. Segmentation results produced from image differencing between current frame and a reference image produce better results compared to subtraction between consecutive frames as this type of subtraction may lead to the generation of false positives where dark shadows move away from an area of background. The method produces a sequence of images whose running median is the reference image. The value of each pixel in the reference image is increased by 1 if the corresponding pixel value in the current image is greater and the value of each pixel in the reference image is decreased by 1 if the corresponding pixel value in the current image is less. Each pixel in the reference image then converges

to a value for which half of the updated values are greater and half are less which actually indicates the median. Among the different advantages of this method, one is that it is computationally inexpensive as it needs to store only one reference image. Moreover, the median possesses better capability of rejecting outliers than the mean in the distribution of values of pixels. Due to these several attributes, we have used this method for calculating foreground pixels in our DTV framework and for the UCSD dataset in UIOC framework.

Another remarkable method of background subtraction is the mixture of Gaussians method (MG) [48]. This method initially models each pixel of an image as a mixture of Gaussians by using an online approximation for updating the model. Then it evaluates Gaussian distributions of the adaptive mixture model and determines which is most probable to be a result from a background process. Each pixel is classified based on the Gaussian distribution it represents. If the Gaussian distribution which represents a pixel most effectively, is part of a background model, then the pixel is classified as background. This method works in real time and produces stable results. It has the ability to handle lighting changes, long term scene changes, shadows, repetitive motions from clutter and many other challenges. So, it can be successfully applied to FUDAN dataset which has lighting changes as well as shadows.

Finally, a very recent and fascinating method of background subtraction needs to be mentioned here which is Visual Background Extractor (ViBe) [2]. This is a very fast method which outperforms many current techniques in terms of computation speed and segmentation accuracies. The method stores a set of values for each pixel acquired from the past in a similar location or in its neighbourhood. Then it compares this set of values with the present pixel value for determining whether the pixel corresponds to the background or not. Then the model is updated by randomly selecting a value for replacing from the background model. If the pixel is classified as part of the background, then its value is propagated in the background model of its neighbouring pixel. This technique is different from the state of art in the sense that it does not work according to the belief that oldest values should be substituted first. We have successfully applied this method on the 12 different videos of the

LHI dataset and also on the LRT dataset.

Chapter 3

Proposed People Counting from Monocular Videos

Based on the literature review presented in the previous section, we come to know that the problem of people counting can be handled in two different ways - either analysing each individual in the crowd differently or work with the entire crowd environment globally.

The detection-tracking framework works on the principle of analysing each individual separately. In these approaches [28, 34, 64], the individual persons are first detected and then they are tracked. The number of tracked trajectories accounts for the estimate of the number of people. Thus, the count is not dependent on individual frames, but on a sequence of frames. The shortcoming of these methods is that, often they avoid a validation step, where the detected objects or tracked trajectories should be classified as a human or non-human. Consequently, these methods are imprecise. Therefore, we develop a framework, which has three major steps: people detection, tracking and validation. The interesting characteristic of the framework is the inclusion of the validation step, which is overlooked by most of the existing methods. Although different methods are proposed for the detection, tracking as well as the validation stages for the framework, the framework is not constrained by these methods. Thus this framework is flexible enough for future work.

The detection-tracking-validation (DTV) technique works well for situations where the object size is large, crowd is not too dense and occlusion is not severe. Large object size helps in the detection as there will be enough number of image

pixels to depict the object. Tracking is failsafe for overhead views where there is no occlusion. In case of whole body views, where there is partial occlusion, particle filter based tracking can be applied.

Applying the detection-tracking-validation approach becomes difficult in dense crowds where each person is depicted by a few image pixels and people occlude each other in complex ways. Detection becomes challenging due to both occlusion and small size of people. Occlusion also poses a difficult challenge for tracking. For these situations, we need to gather all information from the image by analysing the environment globally in order to perform the people counting task successfully. Based on this idea, we develop a framework where we perform the people counting task by monitoring the entire crowd holistically.

In order to design the framework, we adopt a combination of (a) the concept of influx and outflux of fluid mass from computational fluidics, (b) a non-linear regressor to estimate the number of people within a region of interest (ROI) and (c) ROI boundary tracking (as opposed to object or feature tracking) for a short period. Within a ROI, we compute influx/outflux rate, i.e., number of people entering or exiting the ROI per unit time. Then, we sum the influx/outflux rate between any two time points to estimate the number of people that entered and/or left the ROI within that time interval. This framework is named as Unique Influx Outflux Count (UIOC). The UIOC framework is online in nature and is as fast as realtime.

A detailed explanation of the frameworks is given in the ensuing sections.

3.1 Detection-Tracking-Validation (DTV) Framework

In this section we describe the DTV framework [55]. This framework is tested on two major types of datasets - top views and whole body views of people. The input of the proposed algorithm is a sequence of views of passengers and the output is a set of valid trajectories. The number of valid trajectories represents the number of people. The framework contains three major consecutive steps-

i. The object detection step where the trajectories are initiated.

ii. The object tracking step where the trajectories are generated.

iii. The object validation step where the trajectories are classified into human or non-human.

After execution of these three steps, the number of valid trajectories denotes the total number of people in the given sequence of frames. First, the approach for top views is discussed in the following section.

3.1.1 Top Views

The first type of dataset on which the proposed framework is tested, consists of top views of passengers [50] as shown in Figure 3.1(a). The advantage of working with top views is that, there is no occlusion and a person can be tracked failsafe. But the disadvantage is that, there is less number of features that make the automatic detection process challenging. The different stages of working with top views are described below.

Object Detection

Taking into account that there is circularity in the top view of a human body, the whole body of the person has been captured as a circular object. Initially, the frames that do not have any people are removed with an approximate median (AM) based background subtraction method [41] to speed up the process. Canny's edge detector [27] is used to detect edges on the frames which remain after background subtraction. Circles having radii within 60 to 80 are detected on the edge image using Hough circle method [27]. A square template is constructed around the center of the detected circle to denote the object and track it in the following frames Figure 3.1(b). As detection and tracking are performed at each frame, color information is used for distinguishing newly arrived persons and the persons already detected in the previous frame. For each frame, the value of the Bhattacharya coefficient [25] between the color distribution of a newly detected and previously detected person is calculated. If this value becomes very high, then the newly detected person is ignored as he or she is already considered as detected in the previous frame.

Histogram of Oriented Gradients (HOG) [20] is also used for the detection process for comparing with the Hough circle method. In the experiments, as the top views of passengers have a parametric circular shape, the performance of Hough circle method is better than HOG method. Moreover, passengers are having different hair colors, wearing different types of hoodies, caps, long winter jackets, carrying bags etc. These make an irregular shape of the outer body that is very difficult to learn with supervised shape based object detection technique like HOG.

Object Tracking

After an object is detected in a frame, Horn-Schunck (HS) optical flow (OF) method [29] is used to track the center of the object along with its template in the consecutive frames. In the OF based tracking method, the average velocity of pixels within a template of the previous frame is calculated. As the template is in the previous frame, its center position is already known. So, the center of the template in the current frame is obtained by adding the average velocity with the center of the template of the previous frame. Thus tracking is done by obtaining the position of the center of the templates in each frame. As a person is detected in a frame, tracking is initiated and it continues as the person moves through the frame. When the person leaves the FOV, tracking is stopped and the trajectory gets generated as shown in Figure 3.1(c). Apart from Horn-Schunck method, other two well-known methods proposed by Lucas-Kanade method [39] and Brox *et al*. [7] are also used for tracking. All the three methods show good performance on the top views of the passengers. Since Horn-Schunck technique has fewer parameters and is also very fast, it is used in our proposed DTV framework.

Object Validation

Two types of false alarms are generated from Hough or HOG based object detection technique : (a) clutter detected as people; (b) duplicates: detecting different body parts of the same person (Figure 3.2). For clutter removal, an approximate median (AM) based background subtraction method [41] is used in the spatio temporal domain and a measure of overlap of two trajectories is calculated for du-

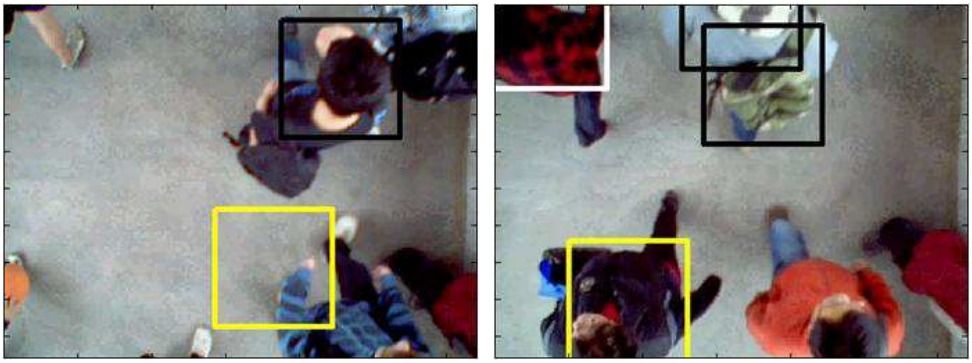
plicate removal. The proposed validation framework is named as spatio-temporal validation (STV). Let $T = T_1, T_2, ..., T_i, ..., T_p$ be a trajectory generated by the tracking algorithm on *p* consecutive frames, where T_i is the set of pixels contained by the trajectory on *i*-th frame. Trajectory T belongs to a person correctly if $\sum_{i=1}^{p} \sum_{(x,y)\epsilon T_i} F_i(x,y) / \sum_{i=1}^{p} N_i > L$. F_i is the output of AM for frame I_i. The value of each pixel (*x, y*) in F_i is either 1 or 0 if it belongs to foreground or background respectively. N_i is the number of pixels contained by the trajectory T_i on i-th frame. So, the proportion of foreground pixels to the total number of pixels in a whole trajectory is calculated to conclude whether the trajectory actually belongs to a person or not as a greater proportion of foreground pixels indicates that the trajectory belongs to a human being. For duplicate removal, one trajectory A is considered as duplicate of another trajectory B if $(\sum_{i=1}^{m} A_i \bigcap B_i / A_i \bigcup B_i)/m > O$ where, *m* is the number of common frames for both trajectories A and B. The values of *L* and *O* are determined empirically.

(a) Input frame

(b) Detection with square template

(c) Trajectory generated after completion of tracking

Figure 3.1: Top views of passengers



(a) Clutter detected as person (b) Duplicate : A person detected twice

Figure 3.2: Different false alarms on top views of passengers

3.1.2 Whole Body Views

The second type of dataset on which the framework is tested, consists of whole body views of people where the passengers are going down or climbing up the staircase of a LRT station. As the view consists of whole body of the passengers, lots of features are available. But the two major challenges in this situation are occlusion and scaling effect. The scaling effect is due to the fact that the size of the people is decreasing or increasing as they are going down or coming up the stairs. The second challenge, occlusion, is not severe in the dataset, as in most of the cases, people are descending or coming up the stairs one by one. In case of two or more people, there is only partial occlusion. In this work, we attempt to avoid occlusion by creating a ROI at the bottom of the frame as shown in Figure 3.3(a). When one person enters the ROI, he/she is detected. If a second person also comes in along with the first person within the ROI, the second person is not detected initially due to partial occlusion and also because the entire body of the second person does not fit simultaneously with the whole body of the first person within the small ROI. Once the initially detected person is tracked for a few frames and the whole body of the second person fully enters the ROI, the second person is captured Figure 3.3(b). In this way, partial occlusion is avoided. Moreover scaling effect can also be handled in this way as the size of a person does not change much within a small region. The various steps of this methodology are described below.

(a) Region of interest on current frame (b) Partial occlusion handling

Figure 3.3: Whole body views of passengers

Object Detection

In the case of whole body views, background subtraction method is adopted for the object detection process. Initially, foreground pixels are obtained from the background detection process. Different connected components are found out from this foreground image. The connected component having area greater than or equal to an average human size corresponds to a single person Figure 3.4. A rectangular template is constructed around the center of the detected blob to denote the object and track it in the following frames.

Histogram of Oriented Gradients (HOG) method is also used for the detection process for comparing with background subtraction method. But in this case, the performance of background subtraction method is better as HOG produces a lot of false alarms. Moreover, the background subtraction method is much faster than the HOG method as established in the result section.

Object Tracking

After an object has been detected in a frame, HS optical flow method is used to track the center of the object along with its template in the consecutive frames. In the OF based tracking method, the average velocity of pixels within a template of the previous frame is calculated. As the template is in the previous frame, its center position is already known. So, the center of the template in the current frame is obtained by adding the average velocity with the center of the template of the

previous frame. Thus, tracking is done by obtaining the position of the center of the templates in each frame. As a person is detected in a frame, tracking is initiated and it continues as the person moves through the rectangular ROI. When the person leaves the ROI, tracking is stopped and the trajectory is generated.

Object Validation

The false alarms are again of two types for both the background subtraction and HOG detection techniques (a) clutter detected as people; (b) duplicates: detecting different body parts of the same person Figure 3.5. For clutter removal, a motion histogram based technique [21] is adopted to classify the trajectories into human and non-human. At first, half of the number of frames is taken for the training process. In the training phase, the motion histograms of all the trajectories are constructed and a support vector machine (SVM) [19] is trained with these histograms. In the testing phase, the motion histograms of the generated trajectories are constructed and using these histograms and the trained SVM, the trajectories are analysed to classify them as human or non-human. For duplicate removal, one trajectory A is considered as duplicate of another trajectory B if they have some overlap between them as described in section 3.1.1.

(a) Input frame (b) Output of background subtraction method

Figure 3.4: Detection algorithm on whole body views

(a) Clutter detected as person (b) Duplicate : A person detected twice

Figure 3.5: Different false alarms on whole body views of passengers

3.2 Results of DTV Framework

The DTV framework is tested on 7000 frames of top views and 10000 frames of whole body views of the LRT dataset. For top views, comparisons among different methods for each of the detection, tracking and validation steps are demonstrated. In the case of top views, the video has different crowd densities whereas for the whole body views, the crowd density is almost constant.

In addition to the LRT dataset, the framework is also tested on 3300 frames of an university campus (LHI) dataset which consists of top views of students walking along the campus.

3.2.1 Results for Top Views

Some visual results of the algorithm on top view sequences for both sparse and dense crowds of the railway dataset, are shown in Figure 3.6. Figure 3.7 demonstrates the visual results of HOG and Hough based detection algorithms. The quantitative comparison in terms of Accuracy, Recall, Precision and F-measure between HOG and Hough are shown in Figure 3.8. In case of detection with HOG, first 50% of the total frames has been used for training and the remaining 50% has been used for testing. F-measure combines recall and precision into a single quantity by computing harmonic mean of Recall and Precision. A better performance is indicated by a higher F-measure. It can be concluded from both Figures 3.7 and 3.8

that Hough circle method performs better than the HOG method. While performing the performance evaluations, Horn-Schunck method is used for tracking and STV method is used for validation for both HOG and Hough based detection procedures.

In the Hough circle method, there is a noise sensitive parameter. This parameter is the ratio of the number of detected edge pixels to the calculated perimeter of the circle. Recall and precision are computed on 100 randomly selected images for different values of this parameter ranging from 0.2 to 0.9 at an interval of 0.1 and it is found that the recall is 100% but precision is 70% at a threshold value of 0.7. This experiment is shown in Table 3.1. The threshold value 0.7 is chosen while performing detection with Hough circle method as the recall value is maximum in this case and the precision value increases significantly by introducing the validation step into spatio-temporal domain.

Horn-Schunck method is chosen for tracking in the proposed framework as it has only one tuning parameter. Average time taken in seconds to track a person between two consecutive frames of resolution 480-by-640 by Horn-Schunck method is 1.32 seconds.

The validation step, which is the noteworthy novelty of the algorithm, enhances the performance of the proposed method. Before validation, the recall and precision of the system was 100% and 70% respectively, which was analyzed on a frame by frame basis. After validating the entire trajectories through the spatio-temporal validation process, the recall and precision of the system is 97% and 92% respectively. The values of L and O, for clutter and duplicate trajectory removal are chosen to be 0.6 and 0.1 experimentally.

The proposed validation technique has been compared with two other validation techniques viz., motion histogram (MH) [21] and spatio-temporal gradient histogram (STG) [31] and their comparisons are illustrated in Figure 3.9, where it shows that the proposed validation technique outperforms both the MH and STG techniques. Both MH and STG are supervised validation algorithms and first 50% of the total frames is used as training and the remaining 50% is used as testing. Here, it is to be kept in mind that Hough circle method is used as detection and Horn-Schunck algorithm is used as tracking while evaluating the performances of

Threshold	Recall	Precision
0.9	0.9	0.89
0.8	0.96	0.82
0.7	1	0.7
0.6	0.91	0.69
0.5	0.91	0.63
0.4	0.91	0.56
0.3	0.91	0.56

Table 3.1: Recall and Precision values for different thresholds of Hough Circle method

STV, MH and STG.

People entering from any of the four borders of the frame and moving in any direction can be detected and tracked successfully using the proposed framework as shown in Figure 3.10.

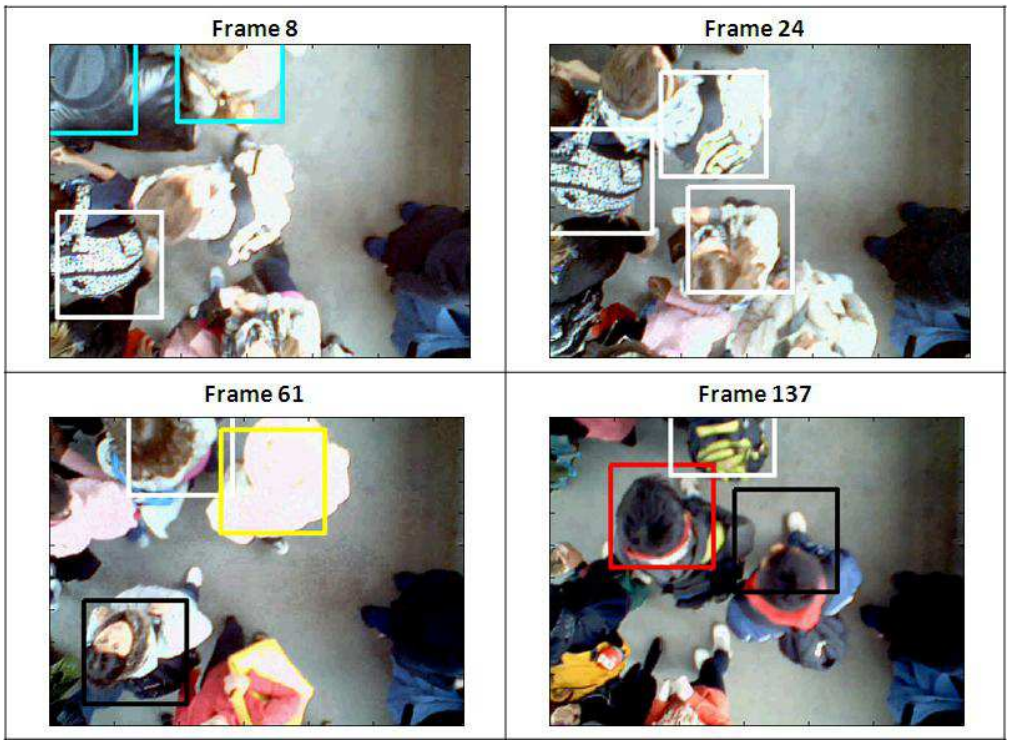
The algorithm also exhibits excellent performance on LHI dataset. Visual results on LHI dataset are shown in Figure 3.11. The video is of duration 5 minutes 30 seconds consisting of 3300 frames where overhead views of passengers are captured. Similarly as the other overhead views of the railway station, Hough circle method is used as detection, Horn-Schunck method is used for tracking and STV is used for validation for the LHI dataset.

Accuracy, Recall, Precision and F-measure of the proposed algorithm for top views on both the LRT and the LHI dataset, are demonstrated in Table 3.2.

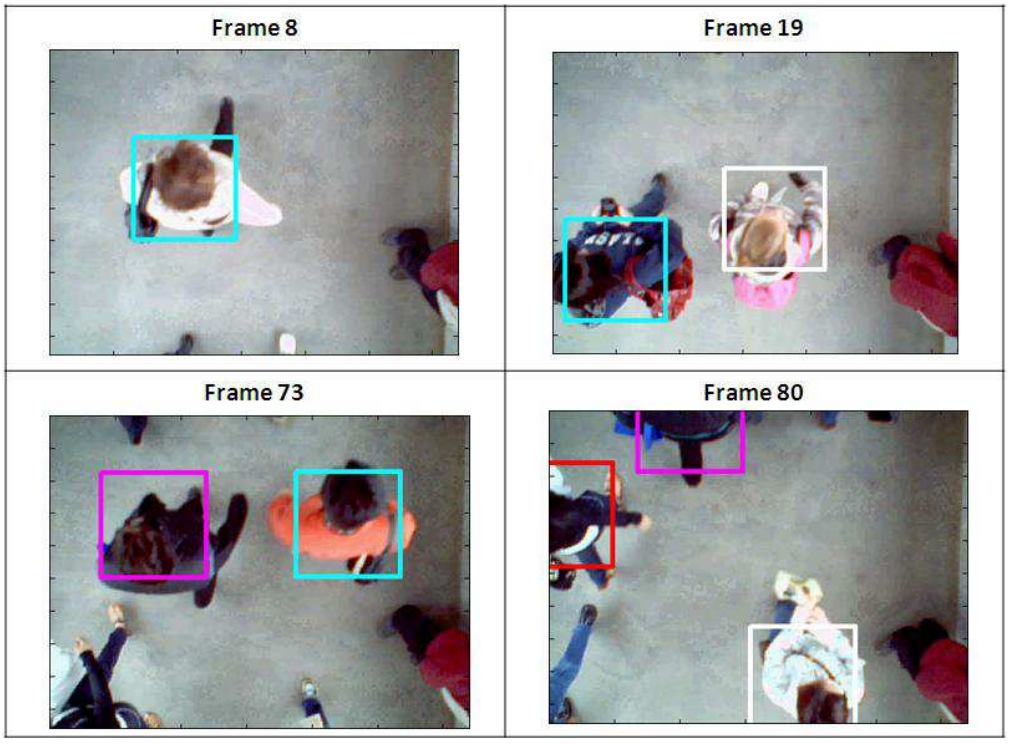
3.2.2 Results for Whole Body Views

The framework is tested on 10000 frames of a video where people were walking down or coming up the stairs of a LRT station. The people were moving mainly in two directions and they wore different types of colored dresses. Some of them were even carrying bags with them. Some visual results are illustrated in Figure 3.12.

Background subtraction method is chosen for the initial segmentation of human



(a) Dense crowd



(b) Sparse crowd

Figure 3.6: Visual results on top views

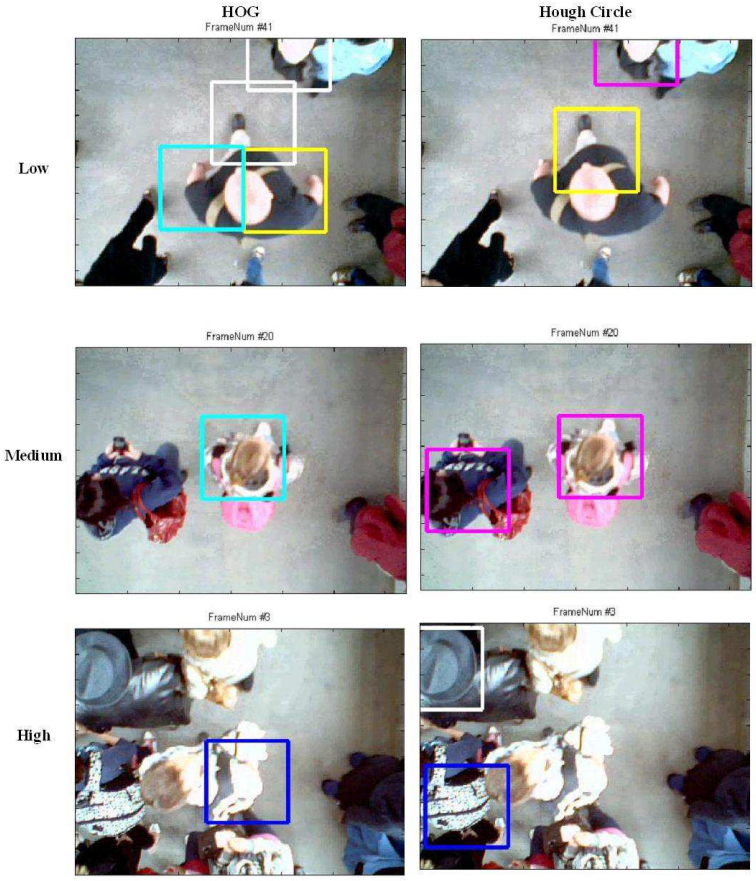


Figure 3.7: Results of HOG and Hough circle based detection

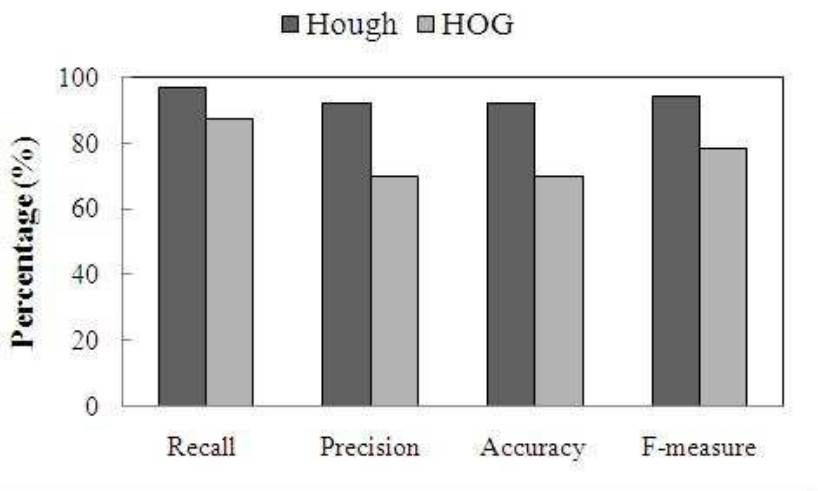


Figure 3.8: Accuracy, Recall, Precision and F-measure for two detection methods: HOG and Hough

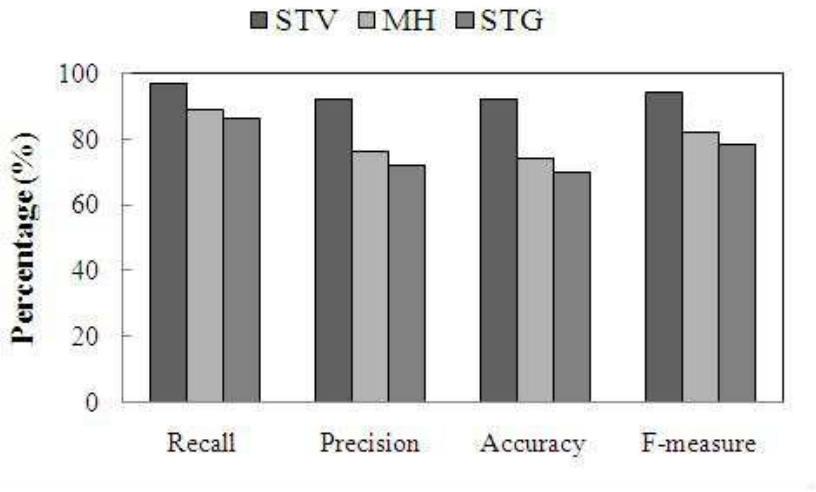


Figure 3.9: Accuracy, Recall, Precision and F-measure for three spatio-temporal based validation methods: motion histogram (MH), spatio-temporal gradient (STG) and proposed spatio-temporal validation (STV)

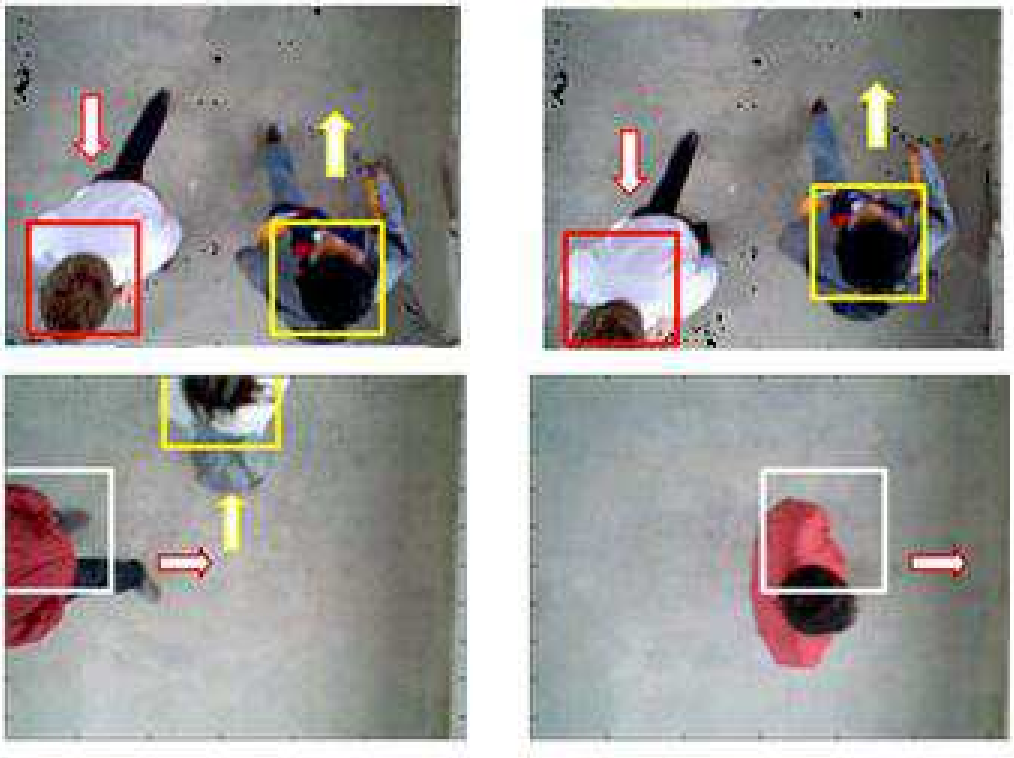


Figure 3.10: People moving in different directions. Top row shows people moving in opposite directions and bottom row shows people moving in perpendicular directions

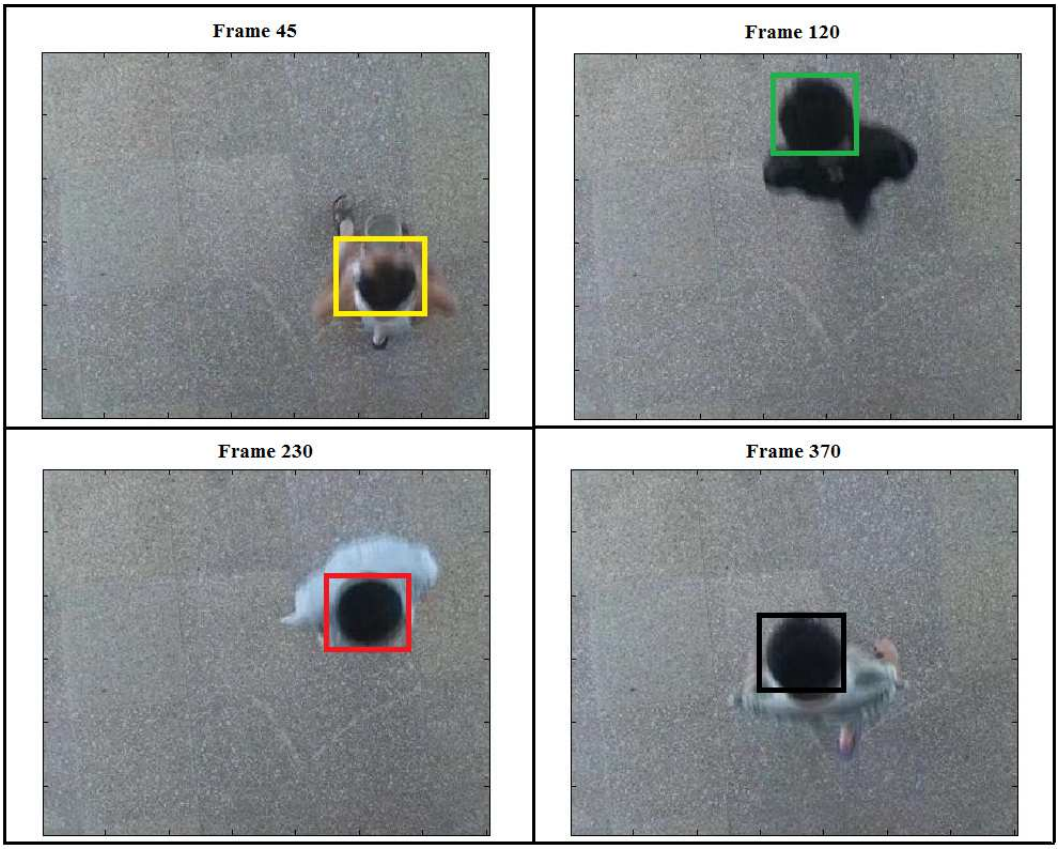


Figure 3.11: Visual results on LHI dataset

Views	**Frames**	**Recall** (%)	**Precision** (%)	**Accuracy** (%)	**F-measure** (%)
Top (LRT)	7000	97	92	92	94
Top (LHI)	3300	99	96	96	97
Whole body (LRT)	10000	95	90	90	92

Table 3.2: Performance of proposed framework

beings. Quantitative comparison between Background Subtraction and HOG based detection methods in terms of Recall, Precision, Accuracy and F-measure is shown in Figure 3.13. Receiver Operating System (ROC) curves for both the detection methods for different ROI's have been plotted in Figure 3.14. The height of the rectangular ROI is varied several times and different values of True Positive Rates (TPR) and False Positve Rates (FPR) are observed which generate the points on the ROC curve. It is noticed that the area under the ROC curve for the proposed background subtraction based detection method is 0.83, whereas the area under the curve for the HOG method is 0.74. The greater area under the ROC curve of the proposed detection method demonstrates its superiority over the HOG method. It is also observed that the background subtraction method is faster than HOG method. The time taken by these two detection methods on each frame implemented in Matlab on a desktop (Intel duo 2 core processor, 2GHz and 4 GB RAM) is illustrated in Table 3.3. The proposed validation step described in section 2.2.3 enhances the performance of proposed method. After passing the entire trajectories through validation process, the recall and precision of the system is 95% and 90% respectively. The recall, precision, accuracy and F-measure of the framework on all the 10000 frames are illustrated in Table 3.2 which proves the competency of the framework.

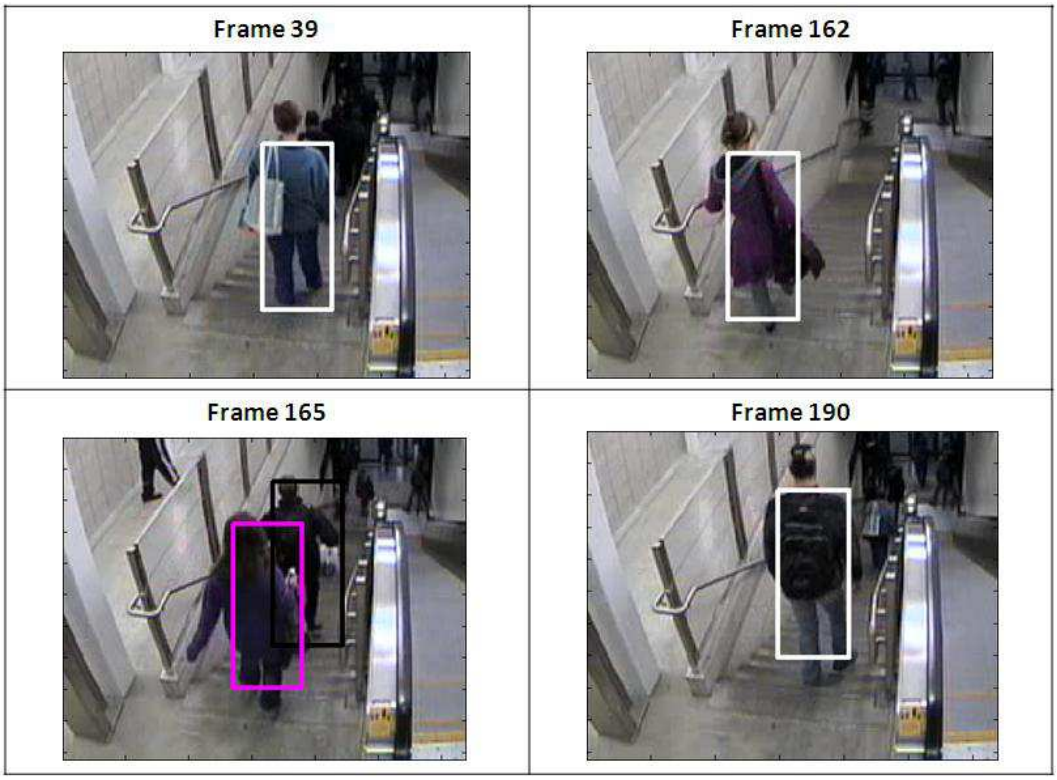


Figure 3.12: Visual results on whole body views

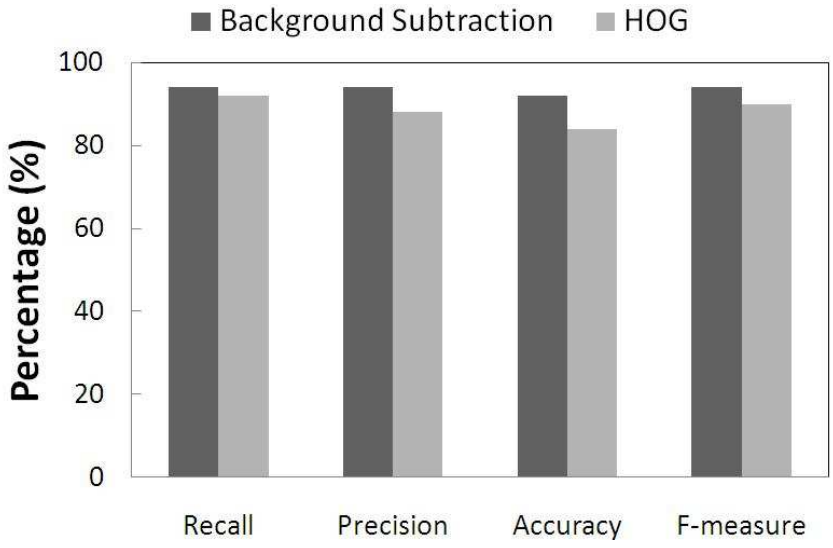


Figure 3.13: Quantitative comparison between Background Subtraction and HOG methods

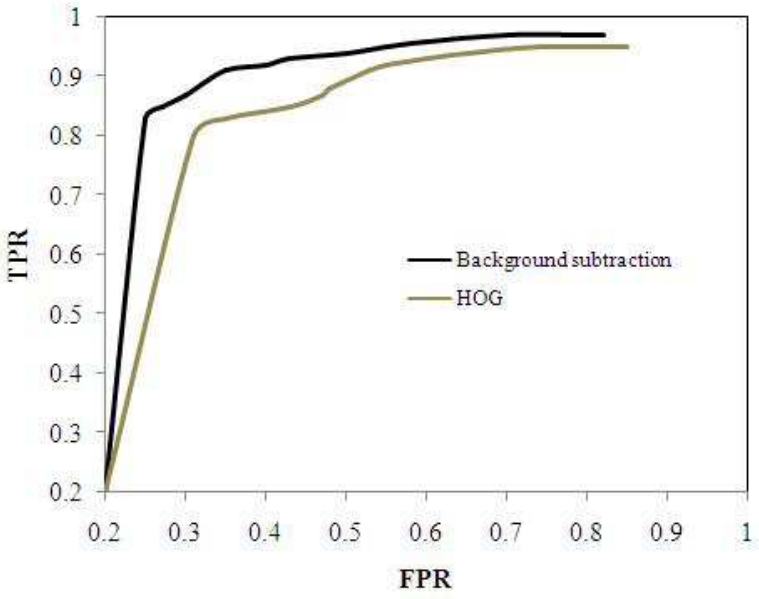


Figure 3.14: ROC curve for comparison between Background Subtraction and HOG methods

Detection Algorithm	Background Subtraction	HOG
Time(s) / frame	0.66	2.76

Table 3.3: Running time of two detection algorithms

3.2.3 Comparison with an existing method

The proposed method is compared with the method proposed by Zeng *et al.* [64] . In Zeng *et al.*'s paper, the approach is almost similar that includes only detection and tracking, but not the validation step. The detection is a supervised method where they use both HOG and Local Binary Pattern (LBP) [56] features to detect the head and shoulders of people to avoid occlusion. In tracking, they use a particle filter tracker. So, basically their approach develops a detection-tracking framework while the proposed approach here designs a detection-tracking-validation framework.

Zeng *et al.*'s method is applied on the same 10000 frames where the proposed framework is implemented. As it is a supervised method, 50% of the total number of frames is used for training and the remaining 50% for testing. As there is no validation step, the false positives cannot not be removed : the accuracy of Zeng *et al.*'s method is 75% whereas the accuracy of the proposed method is 90%. The quantitative comparison of the two methods is illustrated in Figure 3.15 which proves the superiority of the discussed method. ROC curves for both the methods for different ROI's have been plotted in Figure 3.16. The height of the rectangular ROI is varied several times and different values of True Positive Rates (TPR) and False Positve Rates (FPR) are observed which generate the points on the ROC curve. It is noticed that the area under the ROC curve for the proposed DTV framework is 0.83 whereas the area under the ROC curve for Zeng *et al.*'s method is 0.72. The greater area under the ROC curve of the proposed detection method illustrates its superiority over Zeng *et al.*'s method. It is also observed that the DTV framework is faster than Zeng *et al.*'s framework. The time taken by these two methods on each frame implemented in Matlab on a desktop (Intel duo 2 core processor, 2GHz and 4 GB RAM) is illustrated in Table 3.4.

People Counting Algorithm	Proposed Method	Zeng *et al.*'s Method
Time(s) / frame	0.66	2.52

Table 3.4: Running time of two people counting algorithms

Figure 3.15: Quantitative comparison between Proposed method and Zeng *et al.*'s [64] method

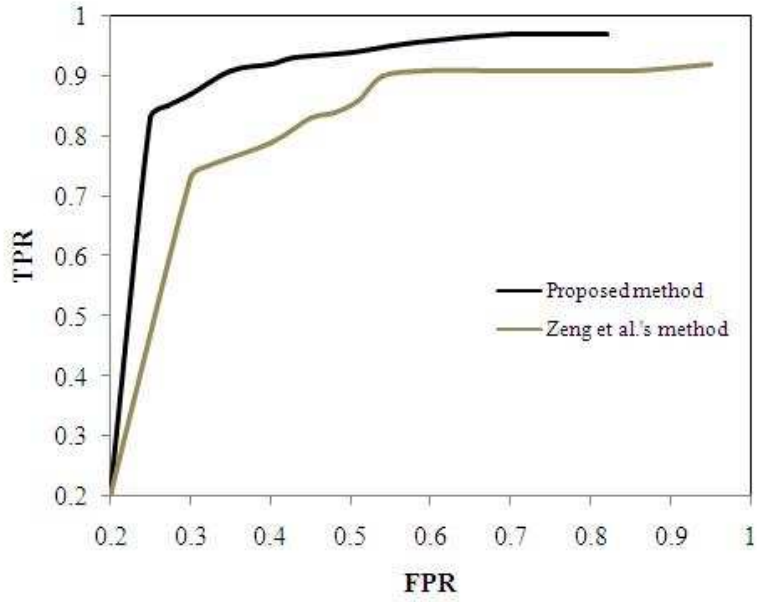


Figure 3.16: ROC curve comparing Proposed method and Zeng *et al.*'s [64] method

The experimental results demonstrated above show that the DTV framework performs well on both types of views and it can successfully detect and track persons having different hair colors, wearing hoodies, caps, long winter jackets, carrying bags and so on. The algorithm also shows promising results for people moving in different directions.

3.3 UIOC Framework

In this section, we will discuss our UIOC framework [53, 52, 51]. This framework counts the unique number of people that entered and exited an ROI within any time interval.

3.3.1 Background of UIOC Framework

At first, we will describe two different techniques that form the backbone of the proposed unique count method. These two techniques are a) frame based count and b) ROI boundary tracking.

(a) Frame Based Count: The general idea here is to extract features from an image frame and map these features to the number of people present in the image frame. This mapping is achieved by supervised machine learning methods, such as Gaussian Process regression [8].

The features taken into account include the foreground features obtained from a background subtraction method and texture features. Based on empirical experiments, the background subtraction algorithms chosen for our framework are the Approximate Median method [41] for the UCSD and the PETS 2009 datasets, Mixture of Gaussians method [48] for the FUDAN dataset and VIBE [2] for the LHI dataset. The features considered for the frame based count are as follows:

i. Segment features are extracted to capture physical properties like shape, size etc. by computing a) foreground area, b) perimeter of foreground area, and c) perimeter-area ratio.

ii. Edge features, such as a) number of edge pixels, and b) edge orientation

are computed. Edges within a segment are strong cues about the number of people in it.

iii. Texture features - Texture features, which are based on the gray-level cooccurrence matrix, are used for estimating the number of pedestrians in each segment [8, 59]. The image is first quantized into eight gray levels and masked by the segment. The joint probability of neighboring pixels i and j within the image frame I, $p(I(i), I(j) \mid \theta)$ is then estimated for four orientations $\theta \in \{0\ , 45\ , 90\ , 135\ \}$.

a Homogeneity: the texture smoothness,

$$g_\theta = \sum_{i,j} p(I(i), I(j) \mid \theta)/(1 + (i - j)^2)$$

b Energy: the total sum-squared energy,

$$e_\theta = \sum_{i,j} p(I(i), I(j) \mid \theta)^2$$

c Entropy: the randomness of the texture distribution,

$$h_\theta = \sum_{i,j} p(I(i), I(j) \mid \theta) log p(I(i), I(j) \mid \theta)$$

Generally, features like foreground segmentation area or number of edge pixels should vary linearly with the number of people in each frame [30, 66]. Foreground segmentation area versus the individual frame-based manual people count over the first 1000 frames of the UCSD dataset is plotted in Figure 3.17. It can be observed that the overall trend is almost linear with some local non-linearities. These local non-linearities occur due to different reasons like occlusion, segmentation errors in background subtraction, perspective foreshortening etc.

The non-linearities are modelled by including additional features, other than the segmentation areas, which are mentioned above and handled by a machine learner using a suitable kernel function. In order to decide our machine learner, we have performed experiments with two non-linear regressors - Gaussian Process (GP) Regressor [58, 73] and Support Vector Regressor (SVR) [47].

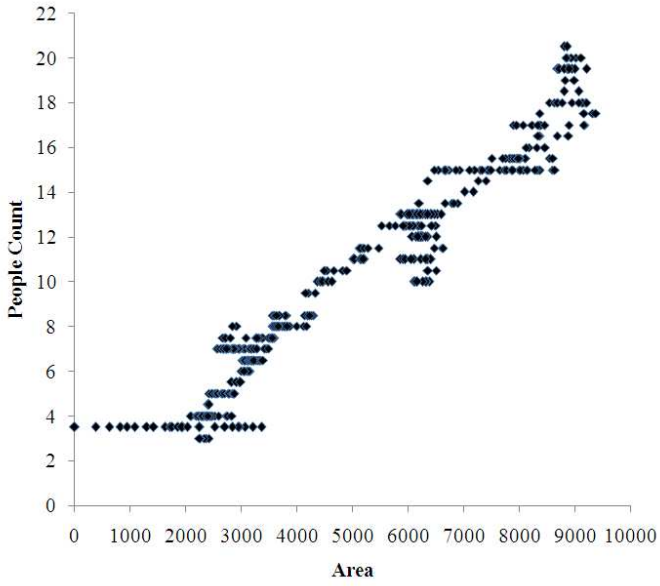
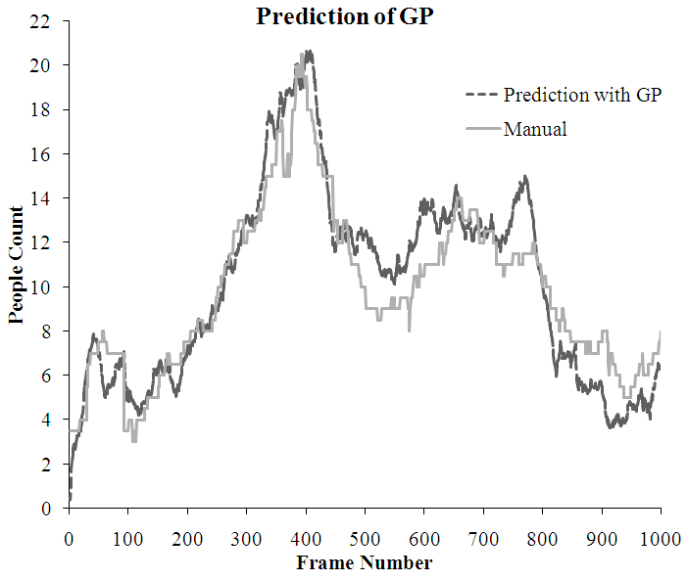


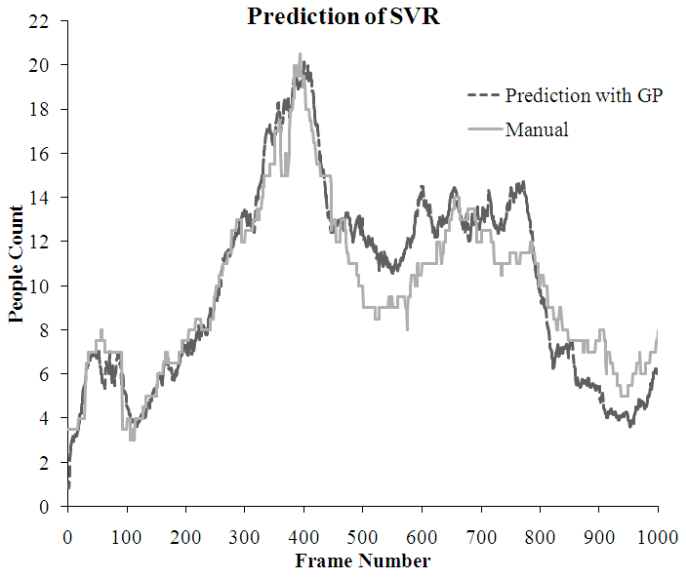
Figure 3.17: Plot of foreground segmentation area vs. people count on first 1000 frames of the UCSD dataset

We choose the UCSD dataset to evaluate the performance of the machine learners as it has many dense crowd instances. For training, the number of people is counted manually on 500 frames with variable crowd densities and the features of each frame within the ROIs are extracted. Next, the machine learners are trained with these extracted features and the corresponding people count in each frame within the ROI to learn the relationship between the two. The performance of the machine learners is then evaluated on 1000 validation frames that are different from the training frames. Manual count is also generated on these 1000 validation frames to perform the quantitative comparison between the two machine learners.

Figure 3.18 plots the predicted count versus the manual count for both the machine learners on the 1000-frame validation set. The dotted lines plot the predicted count from the machine learner, whereas the solid lines denote the true count produced manually. Both the GP Regressor and the SVR perform well on all the frames of the validation set. A quantitative analysis based on mean squared error, mean absolute error and percentage of mean absolute error is reported in Table 3.5. Here it can be seen that the performance of the GP Regressor is slightly better than that of the SVR.



(a)



(b)

Figure 3.18: Performance evaluation of the two machine learners

Machine Learner	Mean Squared Error (No. of people squared/frame)	Mean Absolute Error (No. of people/frame)	Percent Mean Absolute Error (%)
GP	2.3818	1.2378	7.3
SVR	2.5151	1.3001	7.6

Table 3.5: Performance of GP and SVR on 1000 test frames

Based on the above experiments, we have chosen Gaussian Process (GP) Regressor [58] as the machine learner.

A GP specifies distribution over functions. Following the notations from [58], given n dimensional training data,

$$D = \{x^i, f^i | i = 1, 2, ..., n\} = \{X, f\},$$

the key assumption is that,

$$f(x^1), f(x^2), ..., f(x^n) \sim N(O, K)$$

where O is the mean function and K is the covariance function.

We want to make predictions f^* at test points X^*. The joint distribution of f and f^* is Gaussian,

$$\begin{bmatrix} f \\ f^* \end{bmatrix} \sim N(0, \begin{bmatrix} K(X,X) & K(X,X^*) \\ K(X^*,X) & K(X^*,X^*) \end{bmatrix})$$

Here, we are interested in the conditional probability

$$P(f^*|f)$$

which is represented by the posterior,

$$P(f^*|X^*, X, f) \sim N(\mu, \Sigma)$$

where,

$$\mu = K(X, X^*)K(X, X)^{-1}f$$

$$\Sigma = K(X^*, X^*) - K(X, X^*)K(X, X)^{-1}K(X^*, X)$$

Our best estimate for f^* is the mean of the distribution and the uncertainty is captured in the covariance function.

The kernel of the GP used here, is a combination of both linear and squared exponential kernels (RBF) [8]:

$$k(x_p, x_q) = \alpha_1(x_p^T x_q + 1) + \alpha_2 e^{\frac{-\|x_p - x_q\|^2}{\alpha_3}} + \alpha_4 \delta(p, q),$$

where x_p and x_q are the p-th and q-th feature vectors and $\alpha = \{\alpha_1, \alpha_2, \alpha_3, \alpha_4\}$ are the hyperparameters.

Once the machine learner is decided, we test our framework on three background subtraction algorithms to check which one performs the best. Background subtraction is needed to compute the foreground features discussed before. The background subtraction algorithms chosen are Approximate Median method [41], Mixture of Gaussians method [48] and ViBe [2]. For testing these algorithms, the first 1000 frames from the UCSD dataset are used. The unique count of people obtained on the these 1000 frames corresponding to each algorithm are presented in Table 3.6. GP is used as the machine learner in all the three cases. The manual unique count on these 1000 frames is 54. So, it is seen from Table 3.6 that the approximate median method performs the best in terms of accuracy for UCSD dataset. Thus, we choose the approximate median method as our background subtraction algorithm for UCSD dataset. We perform similar experiments for FUDAN and LHI datasets. For FUDAN dataset, we chose Mixture of Gaussians Method and for LHI, we chose ViBe as the best performing background subtraction algorithms for calculating foreground features.

(b) Boundary Tracking with Optical Flow: As has been mentioned earlier, our proposed unique people count is inspired by the control volume analysis in fluidics. Thus, we need to account for people leaving or entering the ROI. To mitigate the effect of occlusion, we avoid the tracking of individual people in our framework. Instead, we track pixels on the ROI boundary over a *short period of time*.

A number of methods can be applied for tracking the ROI boundary. However, we choose a very fast optical flow [29] technique principally to make our framework more suitable for real life applications. The optical flow computes pixel

Background Subtraction Method	Predicted Count	Manual Count	Accuracy (%)
Approximate Median	54.70	52	95.06
Mixture of Gaussians	56.06	52	92.76
ViBe	59.10	52	87.99

Table 3.6: Performance of three background subtraction algorithms on 1000 test frames

motion between two consecutive image frames, taking into account brightness constancy. We have used a publicly available implementation [74] of the method with the default parameter settings in all our experiments. The original ROI and tracked ROI on an image of video 3-3 of the LHI dataset is plotted in Figure 3.19 to cite an example of boundary tracking.

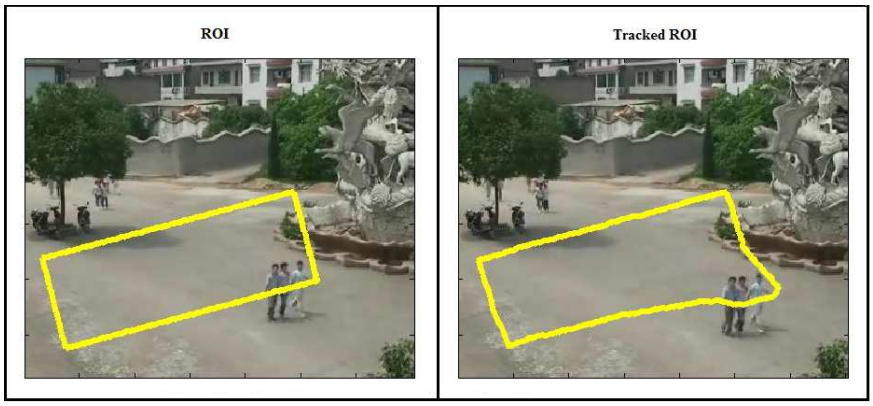


Figure 3.19: Actual ROI and Tracked ROI on an image from video 3-3 of the LHI dataset

3.3.2 Proposed Unique Count Framework

In this section, the proposed framework is presented. The rationale of the framework is based on the assumption of the availability of the following two functional-

ities discussed in the previous section:

Functionality 1: A ROI boundary tracker ($Track$) that is able to track the boundary of ROI R for a short while Δt.

Functionality 2: A machine learner ($Pred$), which is able to predict the number of people present within a ROI on a single video frame.

With these two functionalities, the following framework counts the number of unique people who have entered or left the ROI R.

Unique Influx and Outflux Count (UIOC)

for $t = 0, 1, 2, 3, ...$

$C^t \leftarrow Pred(I^t, R)$;

$R_d \leftarrow Track(I^t, I^{t+\Delta t}, R)$;

$\Delta C_{in} \leftarrow Pred(I^{t+\Delta t}, R \cup R_d) - C^t$;

$\Delta C_{out} \leftarrow C^t - Pred(I^{t+\Delta t}, R \cap R_d)$;

$F_{in}^t \leftarrow \Delta C_{in}/\Delta t$;

$F_{out}^t \leftarrow \Delta C_{out}/\Delta t$;

end

Output at time point t: F_{in}^t, F_{out}^t, C^t.

Unique influx count between t_1 and t_2 is $C^{t_1} + \sum_{t=t_1}^{t_2} F_{in}^t$, and unique outflux count between t_1 and t_2 is $C^{t_2} + \sum_{t=t_1}^{t_2} F_{out}^t$,

where,

I^t: Video frame at time t

R: Region of interest (ROI)

R_d: Deformed ROI due to boundary tracking between frames I^t and $I^{t+\Delta t}$.

ΔC_{in}: Unique influx between time points t and $t + \Delta t$

ΔC_{out}: Unique outflux between time points t and $t + \Delta t$

F_{in}^t: Influx rate of people at time t

F^t_{out}: Outflux rate of people at time t.

The $Track$ functionality tracks the ROI boundary R from I^t through $I^{t+\Delta t}$. $Track$ returns R_d, which is the deformed ROI due to the pixel motion at the boundaries of R. The $Pred$ functionality counts the number of people within a ROI based on extracted image features. If a ROI neither consumes nor generates people, the influx and the outflux count over a period of time should be equal, assuming accurate performance by the two aforementioned functionalities. We refer to such a ROI as a (mass) *conserving* ROI. An example of a non-conserving ROI, where people get consumed and/or generated, is a view of an elevator, in which people enter or come out of.

Figure 3.20 explains the working principle of the framework. The top left part of Figure 3.20, illustrates the positions of people and the ROI R at time instant t. The top right panel displays the positions of people at time instant $t + \Delta t$ as well as the deformed ROI R_d. Notice that R_d is a result of tracking the boundaries of R between t and $t + \Delta t$. The bottom left and right panels respectively show set union and intersection of the original ROI R and the deformed ROI R_d. For clarity, the positions of people at time instant $t + \Delta t$ at the bottom two panels are depicted by dots. Note that influx is given by $\Delta C^t_{in} = Pred(I^{t+\Delta t}, R \cup R_d) - Pred(I^t, R) = 4 - 3 = 1$, whereas outflux is given by $\Delta C^t_{out} = Pred(I^t, R) - Pred(I^{t+\Delta t}, R \cap R_d) = 3 - 1 = 2$. The total unique number of people produced by the influx count is $Pred(I^t, R) + \Delta C^t_{in} = 3 + 1 = 4$ and the total outflux count is $Pred(I^{t+\Delta t}, R) + \Delta C^t_{out} = 2 + 2 = 4$. As expected, these two numbers are equal, since the ROI here is a conserving one that neither consumes nor generates people.

The effect of occlusions is mitigated principally because of three reasons: (a) unlike object tracking, our boundary tracker computes pixel motion only on the ROI boundary, thus working with a very small set of pixels and also for a short period of time. So, it is hardly affected by occlusions (b) machine learner-based frame count can handle occlusion to a great extent and (c) any remaining effects of occlusion overlooked by the machine learner are mitigated by averaging the influx and outflux rate over a period of time. Our experiments validate this observation.

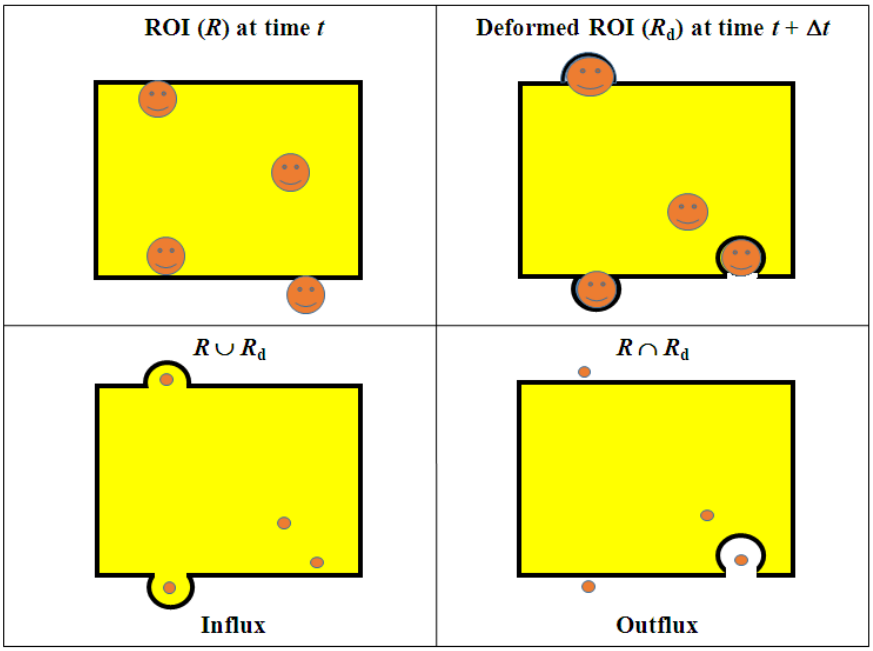


Figure 3.20: Pictorial representation explaining the principal of Influx and Outflux count

According to the above discussion and related to the theoretical discussion of GP in section 3.3.1, we can conclude that the influx (or outflux) rate at time t denoted by F_t^{in} is a random variable and it should follow a Gaussian distribution. Let C_t^0 be the frame basd people count on original ROI and $C_{t+\Delta t}^d$ be the frame based people count on deformed ROI at time $t + \Delta t$. According to the UIOC algorithm in section 3.3.2,

$$F_t^{in} = \frac{1}{\Delta t}(C_t^0 - C_{t+\Delta t}^d) \tag{3.1}$$

Regarding the theory of GP, C_t^0 and $C_{t+\Delta t}^d$ are also Gaussian distributions. Let

$$C_t^0 \sim N(m_t^0, \sigma_t^0)$$

$$C_{t+\Delta t}^d \sim N(m_{t+\Delta t}^d, \sigma_{t+\Delta t}^d)$$

Here m_t^0 and σ_t^0 are mean and standard deviation respectively of the normal distribution followed by C_t^0 whereas m_t^d and σ_t^d are mean and standard deviation respectively of the normal distribution followed by C_t^d.

So according to equation 3.1,

$$F_t^{in} \sim N(\frac{m_t^0 - m_{t+\Delta t}^d}{\Delta t}, \frac{\sqrt{(\sigma_t^0)^2 + (\sigma_{t+\Delta t}^d)^2}}{\Delta t})$$

Let us denote

$$\sigma_{in}^t = \frac{\sqrt{(\sigma_t^0)^2 + (\sigma_{t+\Delta t}^d)^2}}{\Delta t} \tag{3.2}$$

It can be noticed from equation 3.2 that σ_{in}^t should decrease in value with increase of time interval Δt as σ_t^0 does not vary much with time. But as the term $\sigma_{t+\Delta t}^d$ is very much dependent on Δt and may increase in value with increase in Δt value, the value of σ_{in}^t over a long time sequence does not increase much which leads to the competency of our results demonstrated later.

3.4 Results of UIOC Framework

The publicly available monocular videos on which UIOC has been applied are - UCSD, FUDAN and LHI dataset. The UCSD dataset is a 1 hour video containing 25,656 frames. It has captured video of pedestrians passing through University of California, San Diego walkways from a stationary camera. The dimensions of all the videos are 238x158 and captured at 10 fps (frames per second). The dataset is split into 6 scenes captured from different viewpoints. It contains instances of all types of crowd densities - sparse, medium and dense. The number of people on each frame varies from 11 to 45 [59].

The FUDAN dataset consists of 1500 sequential frames, each of dimension 320x240. The video is captured inside FUDAN university campus. The number of pedestrians on each frame varies from 0 to 15 [59]. The dataset has varying crowd density with occlusion, people moving in varying directions and shadows under the pedestrians' feet which make the dataset challenging enough to work with.

The LHI dataset contains 12 videos of Lotus Hill Institute Campus. The videos are captured with camera angles of 90, 65 and 40 degrees respectively. There are 4 videos corresponding to each camera angle and having different views and lengths. As illustrated in Figure 3.23, the first row of images is captured at 90 degree camera angle named from 1-1 to 1-4, the second row corresponds to the 65 degree category,

named from 2-1 to 2-4 and the third row images are taken at 40 degree tilt angle named from 3-1 to 3-4. The scenes covered with the different camera angles vary widely from sparse crowd to dense crowds. The dimension of each image frame is 352x258. The video lengths vary from 3:45 minutes to 25:35 minutes.

Visual results for both the UCSD and FUDAN datasets are shown in Figure 3.21 and 3.22. We can see that both the images have severe occlusions along with shadows of people in FUDAN dataset. For both UCSD and FUDAN datasets, a rectangular ROI R is chosen which can be seen on the top left panels of both the figures, while the top right panels show the deformed ROI R_d. Figure 3.21 shows deformed ROI R_d due to influx while Figure 3.22 shows deformed ROI R_d formed due to outflux. The bottom left panels of Figure 3.21 and 3.22 show the gray images formed due to union ($R \cup R_d$) and intersection ($R \cap R_d$) respectively. The bottom right panels show the foreground/background segmentations of the unified and intersected images, respectively.

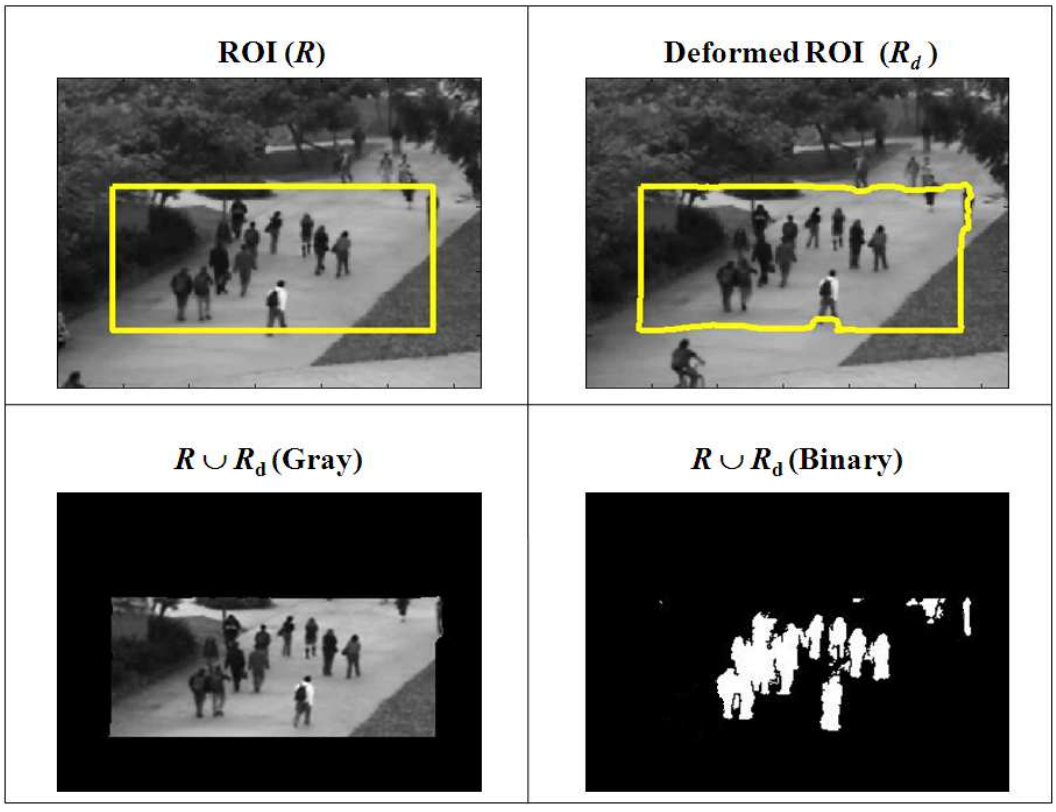


Figure 3.21: Visual results on UCSD dataset for Influx Count

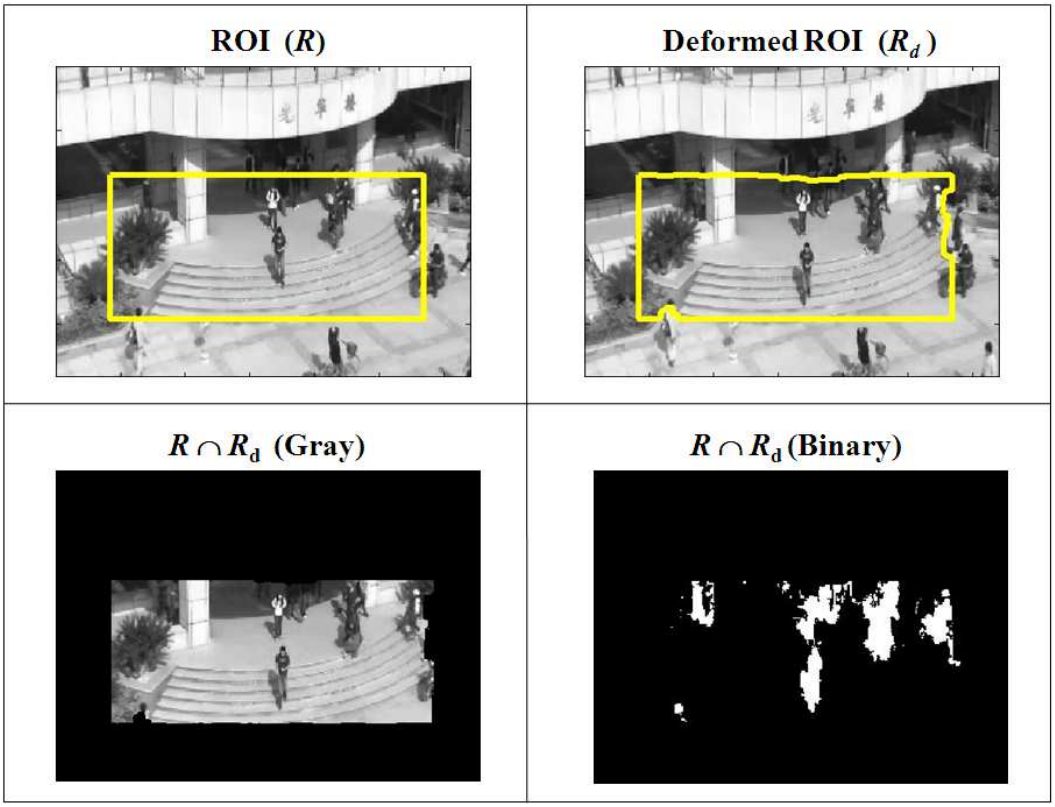


Figure 3.22: Visual results on the FUDAN dataset for Outflux Count

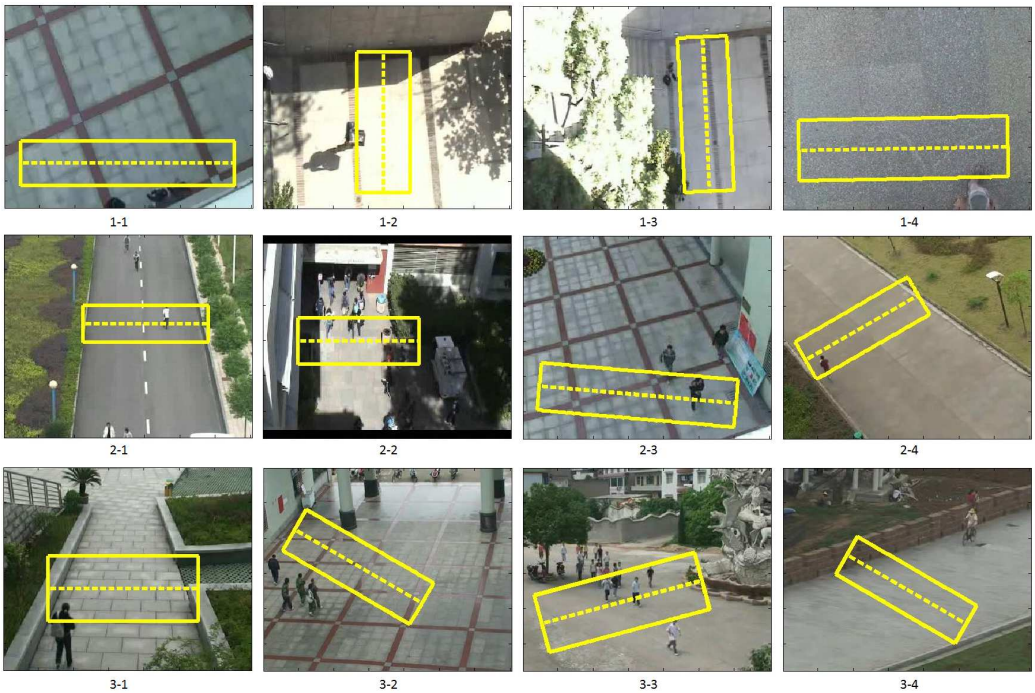


Figure 3.23: Different videos of the LHI dataset. The dotted lines are the LOIs from [16]. The rectangles are our ROIs.

Δt (No. of Frames)	Accuracy (%)
20	91.35
25	98.46
30	93.22

Table 3.7: Accuracy for three different timesteps for the FUDAN dataset

In our method we have only one tunable parameter, the time-step Δt. On one hand, a large Δt would smooth out noisy predictions by the machine learner. On the other hand, a large Δt would make the boundary tracking more challenging due to occlusions. The timestep used for the application of the tracking routine varies for different datasets. These values are chosen based on our experiments with three different values on the first 100 frames. The experiments for the FUDAN dataset are shown in Table 3.7.

In order to evaluate the performance of the proposed algorithm, both the influx and outflux counts are calculated for UCSD and FUDAN datasets, and their means are used as the predicted people count. The values of the influx and outflux count are found to be almost similar for both the datasets. Thus only influx count is computed for the LHI dataset. The performance evaluation of the methods is done by calculating the accuracy as follows:

$$Accuracy = 1 - \frac{\mid ManualCount - PredictedCount \mid}{ManualCount}$$

The manual count, predicted count and the accuracy of the methods on different datasets are tabulated in Table 3.8, 3.9 and 3.10.

The results show that UIOC method outperforms all other methods that are discussed later. The accuracy of people count in UCSD dataset is 94.70 % (Table 3.8), while for FUDAN is as high as 98.46 % (Table 3.9). For LHI dataset, the accuracy of our method remains consistently above 90% (Table 3.10) for most of the videos irrespective of the camera angle.

UIOC works as fast as 30 frames per second on a system with Intel(R), core(TM), DuO CPU, E8400 @ 3GHz. The system is implemented in openCV using the MATLAB implementation of the GP. So this method can be used in real time commercial applications like surveillance videos, transit passenger count in railway stations, road intersections etc.

Performance of UIOC is compared with three methods: a baseline method, detection-tracking method proposed by Zeng *et al.* [64] and flow-mosaicking method [16], which is a LOI counting method. These methods are described next.

3.4.1 Comparison with a Baseline Method

We have devised a baseline method and compared it with UIOC method. The purpose of this method is to prove that unique count of people cannot be computed by direct correlation of frame based count to average foreground pixel speed and distance between the typical entry and exit point of a person on the ROI border. Suppose, we know the average number of frames n_t for which a person is inside a ROI R on frame t. Then, a baseline estimate of the unique people count can be computed between two time points t_1 and t_2 as: $\sum_{t=t_1}^{t_2} Pred(I^t, R)/n_t$, where, as before, $Pred(I^t, R)$ predicts the number of people on frame I^t within the ROI R. A practical and quick approximation to n_t can be obtained by dividing the distance d between a typical entry and exit point on the ROI border by the average foreground pixel speed s_t (obtained by optical flow) computed on frame t. With these approximations, the baseline method people count turns into the formula: $(\sum_{t=t_1}^{t_2} s_t Pred(I^t, R))/d$. Furthermore, we treat the distance d as a tunable parameter here. So, we choose d by matching the baseline count with the manual count on a training set of the first 500 frames.

We apply the baseline method on both the UCSD and FUDAN datset. The total unique count produced by this method for the datasets are 1324.19 and 121.77 respectively as shown in Table 3.8 and Table 3.9. The results also show that our method (UIOC) outperforms the baseline method by approximately 20% and 63% for UCSD and FUDAN dataset respectively. This proves that generating unique count of people from frame based count using direct correlation of frame based

count to average foreground pixel speed and distance between the typical entry and exit point of a person on the ROI border does not provide an accurate people count.

3.4.2 Comparison with A Detection-Tracking Method

In Zeng *et al.*'s work, each individual person is detected in a frame and then tracked in consecutive frames until the person leaves the FOV [64]. The trajectory generated due to tracking represents a single individual. The number of trajectories denotes the number of people during a time interval.

The detection here is a supervised method in which Zeng *et al.* use both Histogram of Gradients (HOG) [20] and Local Binary Pattern (LBP) [56] features to detect the head and shoulders of people to avoid partial occlusion. For tracking, they use a particle filter tracker [12].

Zeng *et al.*'s method is also applied on both the UCSD dataset and the FUDAN dataset. As it is a supervised method, 50% of the total number of frames is used for training and the remaining 50% for testing. Though the detection process is tried to be made robust by taking into account both HOG and LBP features, the detection performance was observed to be somewhat poor on the datasets used here. This happens mainly because of two reasons. Since the size of human beings is very small in the UCSD dataset, the detection process becomes complicated as there are fewer pixels on a human body to detect it properly. The second issue is the occlusion that plagues both detection and tracking.

The performance evaluations of the detection-tracking algorithm are tabulated in Table 3.8 for the UCSD dataset and in Table 3.9 for the FUDAN dataset showing that UIOC outperforms the detection-tracking algorithm for both the datasets.

To demonstrate the competence of our method in handling occlusion, we choose 5 highly occluded video segments from the UCSD dataset each 1000 frames long and calculate the accuracy of the aforementioned methods on these frames. In Figure 3.24 we plot the accuracy of UIOC, baseline and detection-tracking method compared to the manual count at the end of each segment. The UIOC method outperforms the other two methods in handling occlusion in dense crowds with over 95% accuracy. The baseline method performs comparably well in the first 1000

Algorithm	Predicted People Count	Manual People Count	Accuracy (%)
UIOC	1118.27	1062	94.70
Zeng *et al.*	727	1062	68.46
Baseline	1324.19	1062	75.31

Table 3.8: Accuracy of three algorithms on the UCSD dataset

Algorithm	Predicted People Count	Manual People Count	Accuracy (%)
UIOC	75.14	74	98.46
Zeng *et al.*	21	74	28.38
Baseline	121.77	74	35.45

Table 3.9: Accuracy of three algorithms on the FUDAN dataset

frames, because it was initially trained on the first 500 of these 1000 frames. But its performance deteriorates in the ensuing frames mainly due to occlusion.

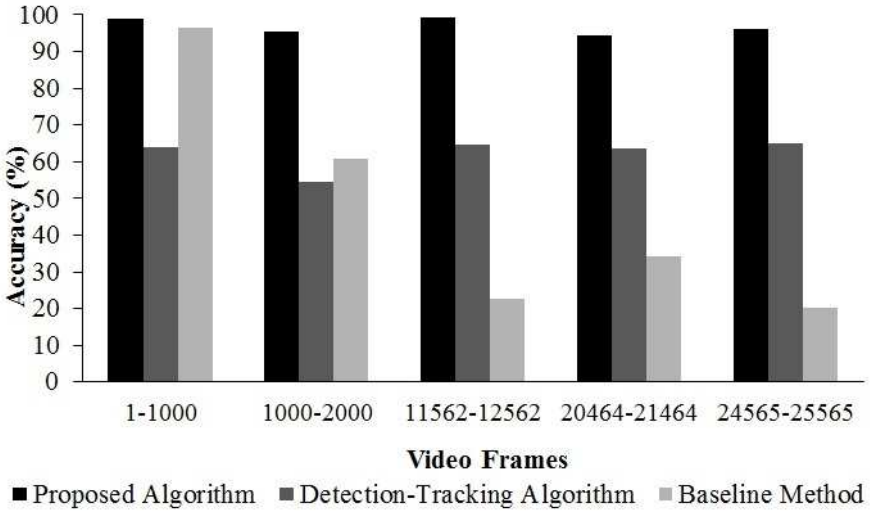


Figure 3.24: Performance evaluation of three algorithms on 5 highly occluded video segments of UCSD dataset

One more important feature of the UIOC method is its ability to avoid error accumulation over the length of a video clip. In Figure 3.25 we wanted to illustrate this phenomena experimentally. In the figure, the accuracy of UIOC method for UCSD dataset is plotted against increasing video clip lengths. The figure shows that the accuracy is barely affected with increase of frame number, thus, confirming our claim.

3.4.3 Comparison with a LOI Counting Method

The LOI counting method described in the Flow-Mosaicking (F-M) paper [16], counts the number of people crossing a specific line of interest based on flow velocity estimation and temporal image generation. The regression function used in the F-M method is the same as ours ie the GP. This method was applied on 12 videos of the LHI dataset [16]. The ROIs chosen on the videos for running our algorithm, are same in locations, dimensions and orientations as that of the ROIs in the F-M method to facilitate fair comparison. These ROIs are shown in Figure 3.23. The accuracy calculated for the unique people count obtained for the videos in LHI dataset

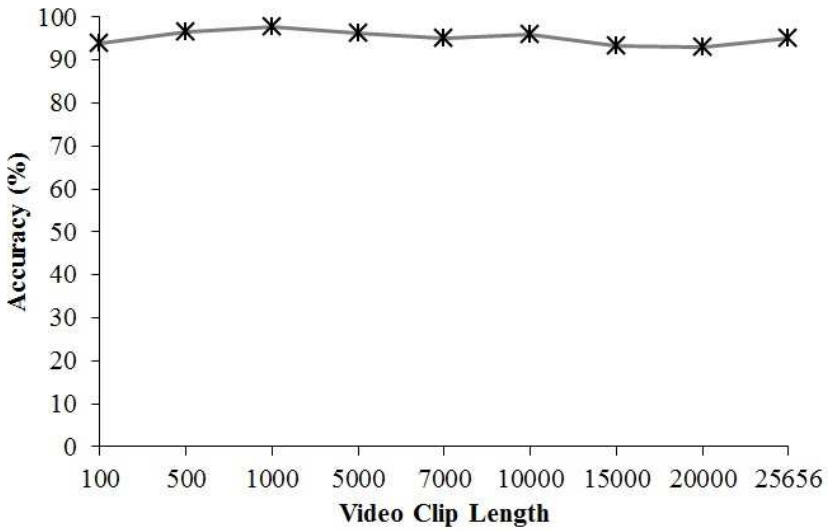


Figure 3.25: Accuracy of the proposed framework with increase of video clip lengths on UCSD dataset

from UIOC and F-M method are listed in Table 3.10 for the sake of comparison. Here, we can observe that UIOC has higher accuracy compared to the F-M method on all the 12 videos. The table also shows the variation of accuracy with change in camera angle for F-M method, with highest average accuracy of 95.24 % at 90 degree or overhead view and lowest accuracy of 82.76% for a video clip at 40 degree camera angle. The results confirm that F-M method using a LOI is incapable of handling occlusion which is absent at 90 degree and generally increases with a decrease in camera angle. On the other hand, the accuracies of our proposed method undergo little variation over different camera angles and views ranging from 91.26-99.72%. Thus we can conclude that the better performance of UIOC over F-M method using LOI is due to effective occlusion handling capacity. The occlusion handling capacity is due to the combination of ROI boundary tracking, using GP with a complex kernel and averaging influx rate of people over time which the LOI counting method lacks in spite of using GP with similar kernel. Another reason for better performance for UIOC method is the dependence of F-M method on temporal image analysis and flow velocity estimation which also suffers from occlusion. We can also surmise that the F-M method with LOI is unsuitable for dataset like the FUDAN where people are moving in random directions, thus complicating the

generation of a single line of interest which will be crossed by every person.

3.4.4 Work on LRT Dataset

As observed from the previous sections, UIOC performs excellent on several publicly available challenging datasets. In this section, I will discuss about its performance on LRT dataset obtained from the City of Edmonton. The dataset is captured from Churchill Square during busy hours of the day. The field of view consists mostly of the platform as shown in Figure 3.27. There are stairs at the top end of the platform from where people are entering the platform. At the two sides of the platform, there are rail lines where trains are coming and leaving. A huge crowd enters and leaves the platform once a trains arrives and leaves. Thus, the motion of the crowd is in multiple directions. It is also observed that there is problem of scaling effect in this dataset because the size of people increases as they descend down the stairs and come nearer to the camera to board the train and their size decreases as they exit the train or enter the platform and then move towards the stairs to exit.

To handle the motion of people in different directions and also the scaling effect, we construct a trapezoidal ROI at the bottom of the stairs as shown in Figure 3.26. It is observed from the video that all the people who enter or leave the platform, cross this ROI. We consider people entering through all the sides of the ROI ie we count the influx count. For background subtraction, we experiment with all the three algorithms - Approximate Median method, Mixture of Gaussians and ViBe. ViBe performs best among all the three. So, we keep ViBe in our framework. For the machine learner we use GP and for optical flow we use Horn Schunck algorithm as before.

The video is of duration 10 minutes with varying crowd densities and the density reaching its maximum when a train arrives. The maximum number of people in a frame was 32. The manual unique count in the video is 123 whereas our algorithm produces a count of 131.86 which is **92.8%** accurate.

Camera Angle	Video name	Video Length (min:sec)	Total no. of pedestrians	Accuracy (%) UIOC method	Accuracy (%) Flow mosaicking method
90	1-1	8:59	256	99.64	97.66
	1-2	14:48	247	97.02	94.33
	1-3	4:30	23	96.61	95.65
	1-4	5:30	180	98.63	93.33
65	2-1	11:29	62	98.27	83.87
	2-2	8:24	300	96.21	84.67
	2-3	3:45	42	91.26	90.48
	2-4	4:40	44	99.72	86.36
40	3-1	7:16	29	97.25	82.76
	3-2	25:35	267	94.64	93.26
	3-3	13:08	288	99.26	93.75
	3-4	10:08	40	93.08	87.50

Table 3.10: Comparative study of the UIOC method and the Flow-Mosaicking method [16] on the LHI dataset

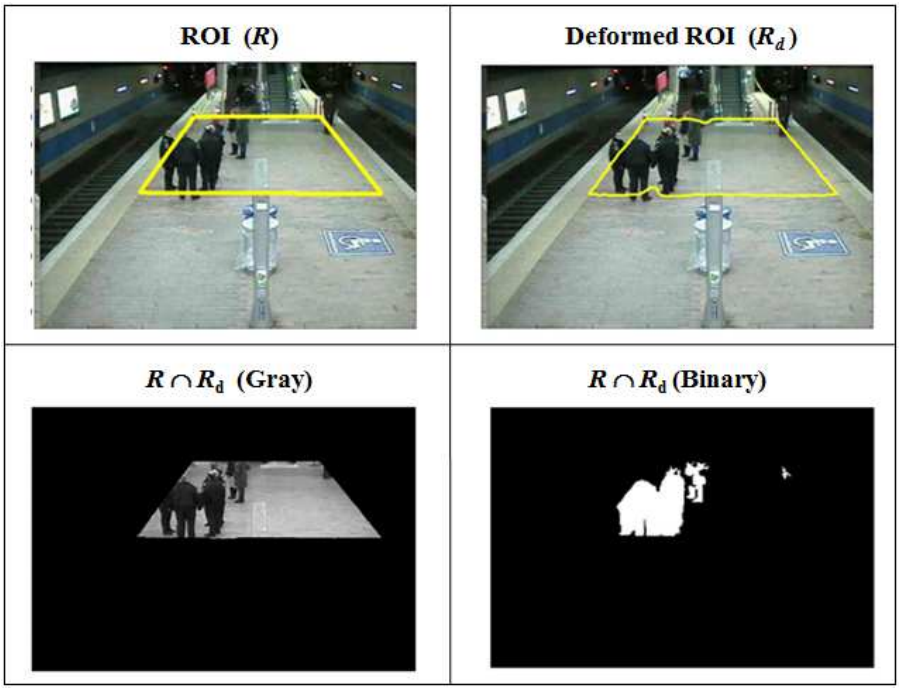


Figure 3.26: Visual Results on LRT dataset for Influx Count

Figure 3.27: Images from LRT Dataset

3.4.5 Work on Multiple ROIs

In order to increase the accuracy of the UIOC framework, we apply it on multiple ROIs, rather than on a single ROI as shown in Figure 3.28. While working with multiple ROIs, we work with a special characteristic of GP. As discussed in section 3.3.1, along with people count estimate in each frame, the GP model also produces the uncertainty in estimation of the people count which is captured in the covariance function of the predictive distribution.

Keeping this in mind, we execute experiments by taking into account multiple ROIs in the following way. -

i We compute the uncertainty in estimation produced by the GP corresponding to each ROI.

ii The ROIs are ranked according to increasing uncertainties.

iii The counts corresponding to the top 3 ROIs are considered.

Once we get the total influx count for all the individual ROIs, we take the average to compute the final unique count. Number of ROIs is a design parameter here. On the training set, we empirically found that we obtained maximum accuracy by working with 5 ROIs choosing the top 3 ROIs among them. The accuracy is **99.12**% on the entire UCSD dataset. In comparison, the unique count was 94.70% with a single ROI previously.

3.5 Addition of Directionality

In addition to total people count, we incorporate directionality in our framework. We test this idea on the UCSD dataset. In the UCSD dataset, the people flow goes mainly in two directions: north and south. In order to count the number of people heading north, we need to take into account the people exiting through the upper boundary i.e., the directional outflux through the upper boundary, because the people who are entering the ROI through the lower boundary are exiting the ROI through the upper boundary. Similarly, for counting the people heading south, we

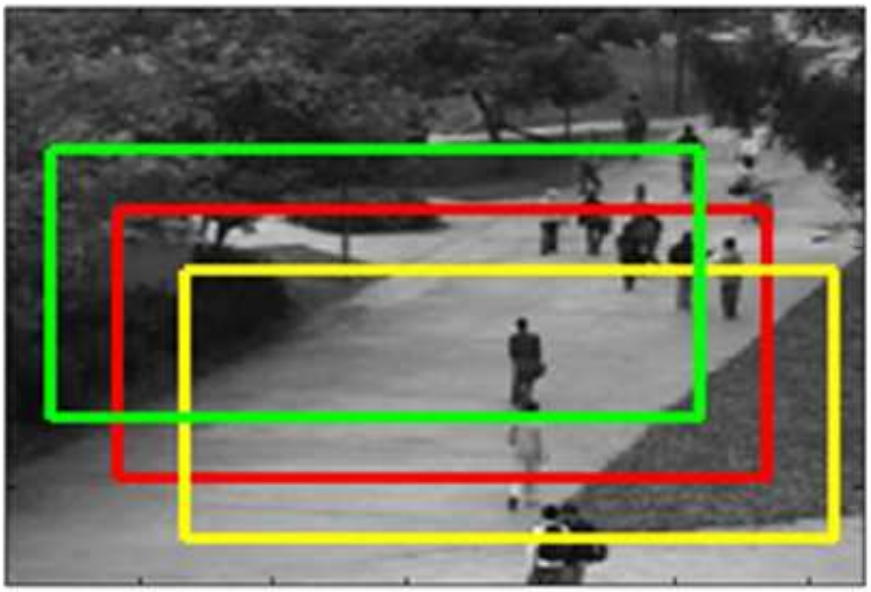


Figure 3.28: Example of multiple ROIs on an image frame of UCSD dataset

need to consider the people exiting through the lower boundary ie the directional outflux through the lower boundary.

Figure 3.29 explains how the directional counting works. The top left panel of Figure 3.29 illustrates the positions of people and the ROI R at time instant t. The top right panel displays the positions of people at time instant $t + \Delta t$ as well as the deformed ROI R_d. R_d is a result of tracking the boundaries of R between t and $t + \Delta t$. The bottom left panel shows R_d intersected with R at the upper boundary, which we need in order to compute the number of people heading north. The bottom right panel shows R_d intersected with R at the lower boundary which we need in order to compute the number of people heading south. The number of people heading north is given by the difference of the number of people present in the actual ROI and the number of people present in the deformed ROI, which is formed from the intersection of R and R_d at the top i.e., $\Delta C_N^t = 4 - 2 = 2$. On the other hand, the number of people heading south is given by the difference of the number of people present in the actual ROI and the number of people present in the deformed ROI, which is formed by the intersection of R and R_d at the bottom i.e., $\Delta C_S^t = 4 - 3 = 1$. Summing ΔC_N^t and ΔC_S^t, we get the total number people moving north and the total number of people moving south respectively.

Direction	Manual People Count	Accuracy (%)
North	183	94.17
South	204	93.23

Table 3.11: Performance of UIOC for Directional Count on UCSD dataset

We test the method on the first video of the UCSD dataset, which has the densest crowd. We manually count the number of people heading north and south separately and then run our framework to get the experimental count. We achieve more than 90% accuracy in both cases as tabulated in Table 3.11.

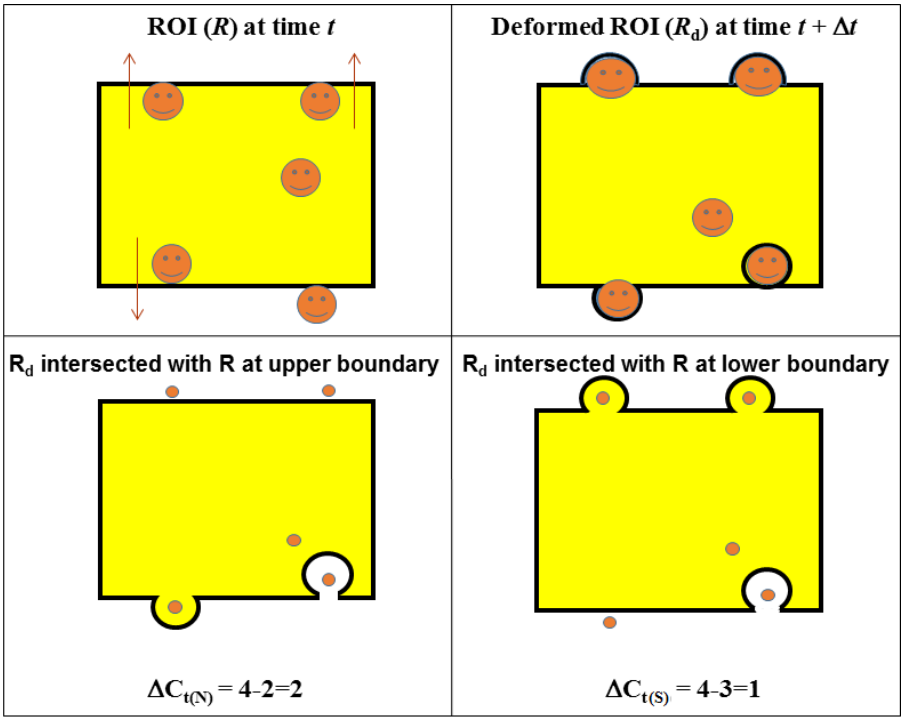


Figure 3.29: Pictorial Explanation of Working Mechanism of Directional Count

3.6 Application on Multiple Views

For extending our framework towards more benchmark datasets, we apply it on multiple views of the PETS 2009 dataset [76] (S1-L2 view, Time 14-31).

In order to apply the UIOC framework on multiple views, the first step is to merge multiple views together in order to choose a ROI. Using a simple program that uses the OpenCV library, the views are merged by their overlapping areas to create an extended view. This is accomplished by manually choosing corresponding points between the source images (views two, three and four) and the destination image (view one) which are the four views presented in Figure 3.30. Using these points and OpenCV library functions, the homography among the views is found and used to transform views two, three, and four into the closest match of view one.

Once the three views are transformed, all four views are superimposed on top of one another for the actual merging. Figure 3.31 shows the merged view. The ROI is then chosen on the merged image. For each view that is transformed, the coordinates of the chosen ROI are transformed using the inverse of the transformation matrix that is used to transform the image to match view one. In this way, the newly transformed ROI corresponds roughly to the correct location on each original view. Also, any points too close to the edges or out of bounds have to be moved in. In the case of the PETS data, as all the views have significant overlap and there is not much room to lose people in, the count for each view should theoretically be almost the same. Therefore, at the end of the program, the average count among all four views is taken as the final estimated people count. The actual count for the selected ROI is 38, and the estimated count is 38.49 which produces 98.71% accuracy.

We compare our results with an existing multi-camera person tracking work [36]. According to [36], the people count accuracy on PETS 2009 S1-L2 dataset (Time 14-31) is almost 82% whereas our accuracy is 98.71% which we achieve without taking into consideration any homography constraints.

So, the UIOC framework, though initially developed for monocular videos, is proved to be flexible enough to perform well even on a network of cameras capturing multiple human views.

From the above experiments, we can observe that the UIOC framework is very well capable of overcoming occlusion, which is one of the most dominant problems in the domain of computer vision based solutions to people counting and also was a shortcoming for DTV method. We achieve more than 95% accuracy on numerous

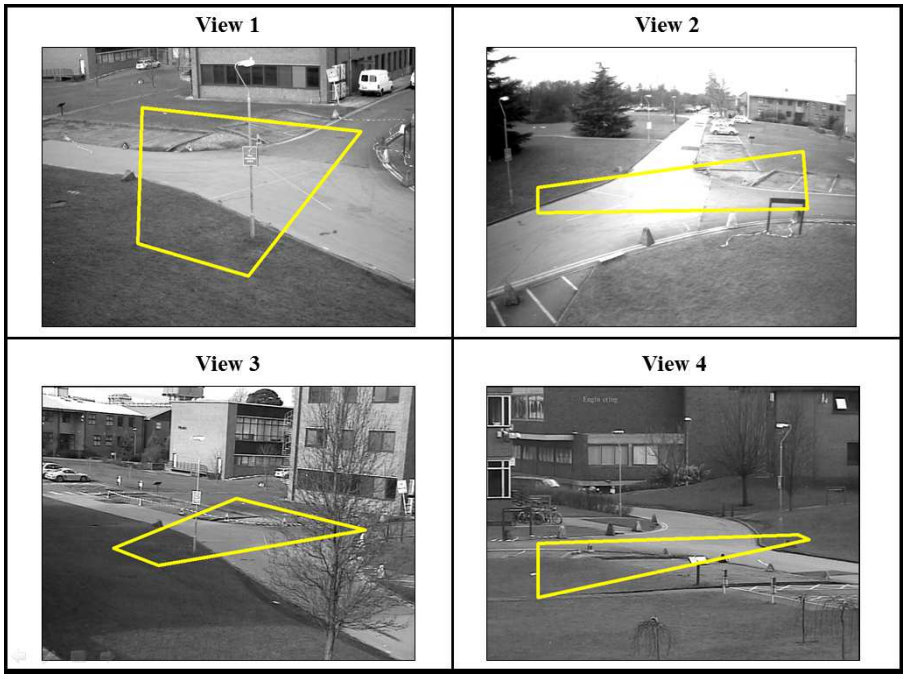


Figure 3.30: The four different views and the chosen ROIs on the PETS 2009 S1-L2 dataset

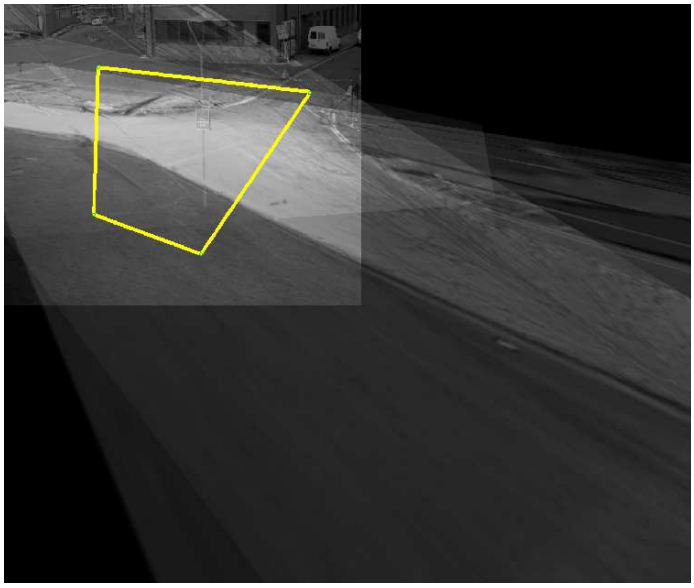


Figure 3.31: Merged view of PETS dataset

publicly available benchmark videos. Apart from producing high accuracy, the method is also online in nature and works very fast which is useful for real time commercial applications. We even extend our framework to work on multiple views with complacent accuracy. In the ensuing chapters, we show that this method is also applied on cell counting videos with satisfactory accuracy which proves its generality and ubiquitousness.

Chapter 4

Generality of UIOC Framework

4.1 Application on Cell Counting

In the earlier chapters, we have discussed the competency of UIOC framework on different types of human videos. The videos include numerous kinds of monocular videos having sparse, medium and dense crowd densities and captured from various camera angles. Apart from monocular videos, the framework is also applied on multiple camera views with high accuracy. All these illustrate its ubiquitousness of performance on different types of human datasets.

In this section, we will discuss about its application on cell dataset and thus illustrating its generality in more details.

Automatic cell detection and counting is a subject of interest for the last few years in many biological and pathological studies ranging from blood cell count to studies regarding cell migration and propagation. All medical laboratories generate blood cell count reports on physician's advice, in order to assist the diagnosis of particular ailments of patients. Sometimes, mobile blood cell assays are studied by time lapse analysis to determine cell counts. The conventional methods, which mostly involve manual counting of blood cells, are unreliable and erroneous and may generate unbearable stress on the laboratory technicians. Cell migration studies, which are important for understanding embryonic development, tissue repair, immune system function, and tumor invasion, also need time lapse video analysis to quantify their response to extracellular chemotactic signals [33] and current analytical methods also show similar problems as blood cell counting.

Introduction of video microscopy has helped in the pathological analysis and has developed the need of sophisticated and automatic cell counting tools to study various aspects of cellular behaviour. Also, the analysis using live-cell imaging provides the window for the use of computer vision-based identification, tracking and counting of blood cells using video microscopy.

An essential element of cell counting from video microscopy has been cell tracking. Manual cell tracking approaches can be erroneous and time consuming, especially when numerous cells are needed to be tracked for a long period of time. Thus, computer vision based automated/semi-automated methods are preferred over the manual techniques. The cell tracking methods can be broadly classified into three categories namely sequential tracking, model based tracking and detection based association.

The sequential tracking techniques [46, 37, 68] mainly uses particle filtering based approach for multiple object tracking. But many of these methods often suffer from computational complexity and the problem of scalability. So, the job of tracking and counting numerous cells for a long period of time become complex and time consuming. The theme of model based tracking [15, 22] is creating and updating a model for each target object to be tracked. But often in biological applications, it may be the case that the type of motion may not be known in advance. In these cases, the model based tracking procedures cannot be applied for cell counting. The idea of detection based association approach [14, 43] is to initially segment and locate objects and then associate the short object trajectories among multiple frames. The shortcoming of this approach is that it may suffer from data association problems and it may sometimes require user inputs.

4.1.1 Results

We use UIOC framework in cell counting and thus avoid individual cell tracking by achieving excellent accuracy in counting.

We validate our method on 11 different cell video sequences. These videos consist of human monocytes observed in an in-vitro assay, where the cells are rolling on human P-selectin (data can be downloaded from: [70]).

We have chosen a rectangular ROI R, as shown in the top left panel of Figure 4.1. The top right panel in Figure 4.1 shows the deformed ROI R_d due to boundary pixel motion. The bottom panel of Figure 4.1 shows $R \cap R_d$ of the foreground/background segmented image. We calculate both the influx and the outflux count for the cells and take their average to get the final total count. We achieve a mean accuracy of 99% for the videos. The time taken to process each frame with UIOC (implemented in Matlab) is 0.1853 seconds on a system with Intel(R) core(TM) i7 processor, 2.2 GHz and 8 GB RAM.

The UIOC algorithm has a tuning parameter: time/frame step (Δt) as mentioned in the previous chapters. On the data set 1, we compute accuracies for three different time steps. These are shown in Table 4.1. Based on these accuracies, we fix the value of Δt as 25 for all the video sequences for computing cell count.

UIOC is compared with six different tracking methods to determine its competency (Table 4.2). Table 4.2 shows the cell counting accuracies of the tracking methods, such as, CGC [13], CG [15], SK [44], CV [14], SS [43] and JL [32] for comparison. The CG method is based on model-based detection approach whereas the other methods are based on detection based association approach as discussed in the introduction. The model-based detection approach was chosen for comparison to establish the inadequacy in performance, due to lack of explicit knowledge of the motion model. The cell counting accuracies (obtained from the tracking accuracies) of these tracking methods are computed from CGC [13]. Since the accuracies are reported for cell tracking, we assume that the accuracy for cell counting will be at most the cell tracking accuracy. As we count a large number of cells over a long interval of time, we do not compare our method with any sequential tracking method for counting, as these methods suffer from computational complexity.

On comparison (Table 4.2), we find that UIOC produces a mean accuracy of 99.07% which is more than any other method compared here. Moreover, the standard deviation is the lowest at 0.65 which makes our method more reliable and reproducible than any other method. Our method greatly outperforms CG due to the lack in knowledge of the required motion model for the image sequence. Among the other detection based tracking methods, CV and SS also perform quite poorly

Time step (Δt)	Accuracy (%)
15	97.82
25	98.41
35	98.12

Table 4.1: Time steps versus accuracy on first data set

compared to other methods. High sensitivity of CV to the maximum speed parameter makes it vulnerable to produce inaccurate results, while use of greedy strategies to solve the correspondence problem is the reason for failure of the SS method. Methods proposed by CGC, JL, and SK have comparable accuracies but are still lower than our method.

Figure 4.2 plots the accuracies of the seven methods including our method for each video sequence. It is clear that our method outperforms every other method with more consistent accuracies over the 11 video sequences. Their is a 3.5% increase in accuracy compared to model-based detection method. Also, our method has outperformed CV and SS methods by 18.0 % and 20.9 % respectively. Our method is capable of outperforming these five methods because it only performs ROI boundary tracking and hence is able to overcome most of the difficulties of individual cell tracking.

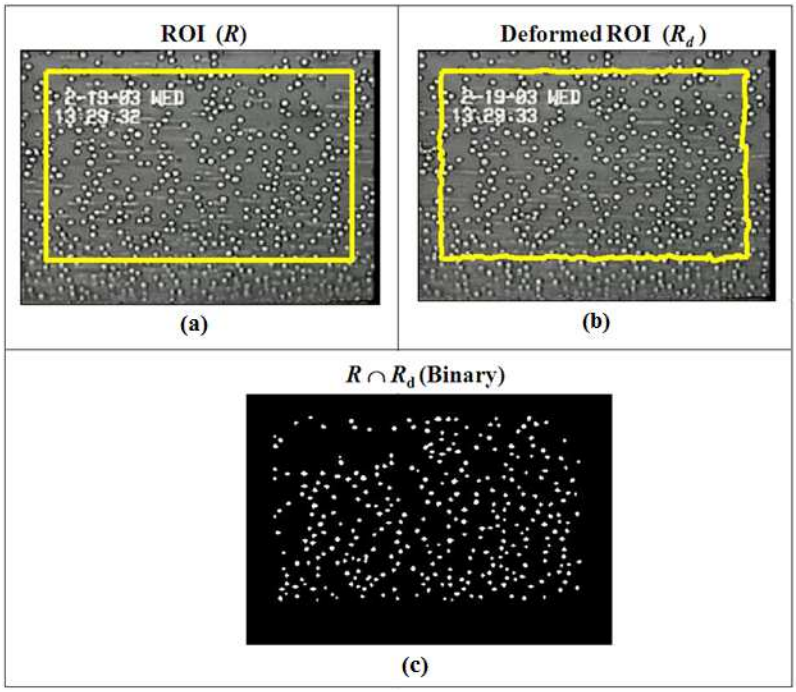


Figure 4.1: Image sequence going through the processing stages. (a) chosen ROI on the cell image from the dataset, (b) deformed ROI due to boundary tracking, (c) intersected foreground/background segmented image

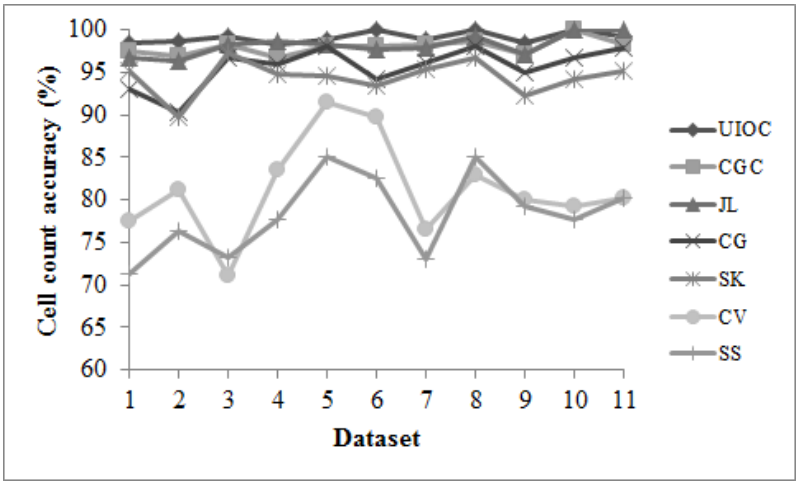


Figure 4.2: Performance graph of the proposed method and other counting methods

Dataset	UIOC	CGC	JL	CG	SK	CV	SS
1	98.41	97.40	96.76	93.07	95.21	77.41	71.26
2	98.56	96.82	96.32	90.35	89.70	81.11	76.30
3	99.19	98.27	98.35	96.72	97.31	71.12	73.27
4	98.27	96.71	98.57	95.85	94.81	83.49	77.62
5	98.88	98.10	98.35	98.10	94.50	91.55	85.03
6	100.00	98.07	97.69	94.23	93.50	89.72	82.62
7	98.81	98.30	97.80	96.22	95.28	76.61	73.13
8	99.91	98.72	99.20	98.11	96.70	82.90	85.10
9	98.52	97.11	97.11	95.00	92.30	80.11	79.20
10	100.00	100.00	100.00	96.70	94.20	79.23	77.65
11	99.23	98.20	100.00	97.91	95.11	80.20	80.20
Mean	99.07	97.97	98.20	95.66	94.42	81.22	78.31
SD	0.65	0.94	1.22	2.39	2.08	5.75	4.70

Table 4.2: Percentage of cell counting accuracy of seven different algorithms for 11 datasets (reproduced from Chatterjee et al. [13])

Chapter 5

Conclusions and Future Work

5.1 Conclusion

In this thesis, we have proposed two frameworks for unique people count from monocular videos - DTV and UIOC.

The DTV framework consists of three components: object detection, object tracking and object validation. A person is detected in the object detection step as he enters the image frame, the object tracking step tracks the person as he moves through the frame and as a result, a trajectory is generated. The validation step analyses the trajectories and the total number of trajectories is counted. The number of valid trajectories represents the number of people. The algorithm is experimented on both top views and whole body views of people. The algorithm succeeds in detecting various types of appearances like persons having different hair colors, wearing hoodies, caps, long winter jackets, carrying bags etc. The proposed framework is also smart enough for handling of people entering in the frame from any direction. Although some specific methods are proposed for the different stages of the framework, the framework is not constrained by these methods. Thus this framework is flexible enough for future work.

The shortcoming of the DTV framework is in handling occlusion. It is competent for overhead views and views having partial occlusions but not competent enough for highly occluded situation.

Our proposed UIOC framework is designed in such a way so that it is capable of handling any type of occlusions and can handle any challenging situations. As

observed from the experimental results, we achieve more than 95% accuracy on numerous publicly available benchmark videos using the UIOC framework. It is also applied on LRT dataset which is a real life data where it has produced satisfactory accuracy. Apart from producing high accuracy, this method is also online in nature and works very fast which is useful for real time commercial applications. We even extend the framework to work on multiple views and on cell counting videos with complacent accuracy. Thus we can conclude that the proposed UIOC method in my thesis is not only competent in nature but also as fast as real time and flexible enough to work on different types of dataset.

Our future work includes the following -

(i) Work on multiple ROIs for LRT dataset.

(ii) Addition of directionality for LRT dataset.

(iii) Application of the framework on vehicle counting.

Chapter 6

Appendix

The optical flow Horn Schunck method has been widely used throughout the thesis. The theory and derivations are given here.

6.1 Estimation of Optical Flow

The optical flow method which is used to calculate motion between two image frames, is described as follows. The motions are computed at every pixel position at times t and $t + \delta t$. We assume that, a pixel at location (x, y, t) with intensity $I(x, y, t)$ moves by $\delta x, \delta y$ within a small time period δt between the two image frames. As we assume the image intensity of the pixel to remain constant, it can be written as [24]:

$$I(x, y, t) = I(x + \delta x, y + \delta y, t + \delta t). \tag{6.1}$$

We assume that the movement of the pixel is small. So, expanding the right hand side of (6.1) with Taylor series, it can be obtained:

$$I(x + \delta x, y + \delta y, t + \delta t) = I(x, y, t) + \frac{\partial I}{\partial x}\delta x + \frac{\partial I}{\partial y}\delta y + \frac{\partial I}{\partial t}\delta t + H.O.T., \tag{6.2}$$

where $H.O.T$ indicates the higher order terms. From (6.1) and (6.2) it is obtained,

$$\frac{\partial I}{\partial x}\frac{\delta x}{\delta t} + \frac{\partial I}{\partial y}\frac{\delta y}{\delta t} + \frac{\partial I}{\partial t} = 0, \tag{6.3}$$

or,

$$I_x\dot{x} + I_y\dot{y} = -I_t, \tag{6.4}$$

where $\dot{x}$ and $\dot{y}$ are the horizontal and the vertical velocities respectively at pixel location (x, y); I_x, I_y and I_t are the derivatives of the image in x, y and time directions respectively. This is an equation having two unknown variables $\dot{x}$ and $\dot{y}$. Thus, it is an under-determined system. This is popularly known as the **aperture problem**. Different optical flow methods generally solve this problem using various regularization techniques, like adding equations arising out of the assumptions about the smoothness of the flow.

6.1.1 Horn-Schunck Method

The Horn-Schunck method computes optical flow by optimizing a functional based on residuals from the brightness constancy constraint and a particular regularization term which denotes the expected smoothness of the flow field [29]. The method is termed as global as it incorporates a global constraint of smoothness for solving the aperture problem.

The algorithm assumes smoothness in the flow over the whole image. Thus, it tries to minimize distortions in flow and tends to provide solutions with more smoothness.

The flow is formulated as a global energy functional minimized thereafter. For two dimensional images the functional is represented as follows -

$$E = \int\int [(I_x u + I_y v + I_t)^2 + \alpha^2(\|\nabla u\|^2 + \|\nabla v\|^2)]dxdy \tag{6.5}$$

where I_x, I_y and I_t denote the derivatives of the image intensity values along the x, y and time dimensions respectively. The OF vector is represented as $\overrightarrow{V} = [u(x, y), v(x, y)]^T$. The parameter α is a regularization constant. Larger values of α results in a smoother flow. This functional is minimized by solving corresponding EulerLagrange equations which are as follows-

$$\frac{\partial L}{\partial u} - \frac{\partial}{\partial x}\frac{\partial L}{\partial u_x} - \frac{\partial}{\partial y}\frac{\partial L}{\partial u_y} = 0 \tag{6.6}$$

$$\frac{\partial L}{\partial v} - \frac{\partial}{\partial x}\frac{\partial L}{\partial v_x} - \frac{\partial}{\partial y}\frac{\partial L}{\partial v_y} = 0 \tag{6.7}$$

L is the integrand of the energy expression which produces -

$$I_x(I_x u + I_y v + I_t) - \alpha^2 \Delta u = 0 \tag{6.8}$$

$$I_y(I_x u + I_y v + I_t) - \alpha^2 \Delta v = 0 \tag{6.9}$$

Here subscripts represent partial differentiation and $\Delta = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$ represents the Laplace operator.

The Laplacian is generally calculated approximately with the use of finite differences and can be written as $\Delta u(x, y) = \overline{u}(x, y) - u(x, y)$ where $\overline{u}(x, y)$ is a weighted average of u which is calculated over the neighbourhood around the pixel assumed to be located at (x, y).

With the help of this notation, the previous equation can be rewritten as follows-

$$(I_x^2 + \alpha^2)u + I_x I_y v = \alpha^2 \overline{u} - I_x I_t \tag{6.10}$$

$$I_x I_y u + (I_y^2 + \alpha^2)v = \alpha^2 \overline{v} - I_y I_t \tag{6.11}$$

This equation is linear in u and v. Thus it can be solved for each pixel in the image.

But, as the solution is dependent on the neighbouring values of the OF field, it should be repeated along with the update of the neighbours.

Keeping this in mind, the following iterations should be operated -

$$u^{k+1} = \overline{u}^k - \frac{I_x(I_x \overline{u}^k + I_y \overline{v}^k + I_t)}{\alpha^2 + I_x^2 + I_y^2} \tag{6.12}$$

$$v^{k+1} = \overline{v}^k - \frac{I_y(I_x \overline{u}^k + I_y \overline{v}^k + I_t)}{\alpha^2 + I_x^2 + I_y^2} \tag{6.13}$$

The superscript $k + 1$ denotes the very following iteration which needs to be calculated and k denotes the previously calculated result.

6.2 Gaussian Process Regressor

The Gaussian Process regressor which is one of the backbones of UIOC algorithm, is briefly discussed here.

A GP specifies distribution over functions. Following the notations from [58], given n dimensional training data,

$$D = \{x^i, f^i | i = 1, 2, ..., n\} = \{X, f\},$$

the key assumption is that,

$$f(x^1), f(x^2), ..., f(x^n) \sim N(O, K)$$

where O is the mean function and K is the covariance function.

We want to make predictions f^* at test points X^*. The joint distribution of f and f^* is Gaussian,

$$\begin{bmatrix} f \\ f^* \end{bmatrix} \sim N(0, \begin{bmatrix} K(X,X) & K(X,X^*) \\ K(X^*,X) & K(X^*,X^*) \end{bmatrix})$$

Here, we are interested in the conditional probability

$$P(f^*|f)$$

which is represented by the posterior,

$$P(f^*|X^*, X, f) \sim N(\mu, \Sigma)$$

where,

$$\mu = K(X, X^*)K(X, X)^{-1} f$$

$$\Sigma = K(X^*, X^*) - K(X, X^*)K(X, X)^{-1}K(X^*, X)$$

Our best estimate for f^* is the mean of the distribution and the uncertainty is captured in the covariance function.

6.3 Background Subtraction

Background subtraction methods are used in the fields of image processing and computer vision for extracting foreground in order to further process it. Generally objects in the foreground like humans, cars, texts etc are the regions of interest in an image. Background subtraction is a popular technique to detect moving objects in videos from static cameras. The rationale behind the approach is that, moving objects are detected with respect to the current frame and a reference frame. This reference frame is known as background model. Background subtraction is mainly needed for images originated from video streams.

A robust background subtraction algorithm should be capable of handling lighting changes, repetitive motions from clutter, long-term scene changes, shadows and many other.

If $V(x, y, t)$ is a video sequence where t is the time dimension, x and y are the pixel locations, then for example, $V(1, 2, 3)$ is the pixel intensity $(1, 2)$ pixel location of the image at $t = 3$ in the video sequence.

6.3.1 Approximate Median Method

In this section, we will discuss the Approximate Median (AM) method of background subtraction which is frequently used in both the DTV and UIOC frameworks.

In order to obtain foreground images using median filter, we adopt the following procedure -

In order to calculate the background image at the instant t

$$B(x, y) = \frac{1}{N} \sum_{i=1}^{N} V(x, y, t - i) \tag{6.14}$$

where N is the number of preceding images whose median is taken into account. N depends on the video frame rate (number of images per second in the video) and the movement amount in image sequences.

$B(x, y)$ is the background.

Once the background $B(x, y)$ is obtained, it can be subtracted from the image $V(x, y, t)$ at time $t = t$ and threshold it. Thus the foreground is

$$|V(x, y, t) - B(x, y)| > \mathrm{Th} \tag{6.15}$$

where Th is threshold.

The background subtraction method often used in this thesis, is the Approximate Median (AM) [41] method. This method is a combined form of image differencing with respect to a median background and a Laplacian operator. The first step in this method is image differencing. Each consecutive image is subtracted from a time averaged reference image. The difference image produced as an output of this step is thresholded. This threshold is the only tunable parameter in the AM method which can be tuned with only a few training frames. Moving object pixels having values more than the threshold value are considered as foreground pixels. Segmentation results produced from image differencing between current frame and a reference image produce better results compared to subtraction between consecutive frames as this type of subtraction may lead to the generation of false positives where dark shadows move away from an area of background. The method produces a sequence of images whose running median is the reference image. The value of each pixel in the reference image is increased by 1 if the corresponding pixel value in the current image is greater and the value of each pixel in the reference image is decreased by 1 if the corresponding pixel value in the current image is less. Each pixel in the reference image then converges to a value for which half of the updated values are greater and half are less which actually indicates the median. Among the different advantages of this method, one is that it is computationally inexpensive as it needs to store only one reference image. Moreover, the median possesses better capability of rejecting outliers than the mean in the distribution of values of pixels. Due to these several attributes, we have used this method for calculating foreground pixels in our DTV framework and for the UCSD dataset in UIOC framework.

Bibliography

[1] M. J. Black and P. Anandan, The robust estimation of multiple motions: parametric and piecewise smooth flow fields, *Computer Vision and Image Understanding,* 63, 75 - 104, 1996.

[2] O. Barnich and M.V.Droogenbroeck, ViBe: A universal background subtraction algorithm for video sequences, *TIP,* 20, 1709 - 1724, 2011.

[3] S. Birchfield, KLT : An implementation of the kanade-lucas-tomasi feature tracker, http://vision.stanford.edu/birch/klt/.

[4] J. Black and T. Ellis, Multi camera image tracking, *Image and Vision Computing,* 24, 1256 - 1267, 2006.

[5] P.C.Box and J.C.Oppenlander, Manual of traffic engineering studies, *Institute of Transportation Engineers,* pp. 17, 2010.

[6] G. J. Brostow and R. Cipolla, Unsupervised Bayesian detection of independent motion in crowds, *CVPR,* pp. 594 –601, 2006.

[7] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert, High Accuracy Optical Flow Estimation Based on a Theory for Warping, *Proceedings of ECCV*, 4, 25 - 36, 2004.

[8] A.B. Chan, Z-S. J. Liang and N. Vasconcelos, Privacy preserving crowd montioring: counting people without people models or tracking, *CVPR,* pp. 1 - 7, 2008.

[9] A.B. Chan and N. Vasconcelos, Counting People With Low-Level Features and Bayesian Regression, *TIP,* 21, 2160 - 2177, 2012.

[10] A. B. Chan and N. Vasconcelos, Bayesian Poisson regression for crowd counting, *ICCV,* pp. 1 7, 2009.

[11] A. B. Chan and N.Vasconcelos, Modeling, clustering, and segmenting video with mixtures of dynamic textures, *IEEE Trans. Pattern Anal.Mach. Intell.*, 30, 909 926, 2008.

[12] T. Chateau, V. Gay-Belille, F. Chausse and J. Lapreste, Real-Time tracking with Classifiers, *ECCV,* pp. 218 - 231, 2006.

[13] R. Chatterjee, M. Ghosh, A.S. Chowdhury and N.Ray, Cell tracking in microscopic video using matching and linking of bipartite graphs, *Comput Methods Programs Biomed*, 2013.

[14] D. Chetverikov and J. Verestoy, Feature point tracking for incomplete trajectories, computing, *Devoted Issue on Digital Image Processing,* 62, 321 - 338, 1999.

[15] J.C. Crocker and D.G. Grier, Methods of digital video microscopyfor colloidal studies, *Journal of Colloid Interface Science,* 179, 298 - 310, 1996.

[16] Y. Cong, H. Gong, S. Zhu, and Y. Tang, Flow mosaicking: Real-time pedestrian counting without scene-specific learning, *CVPR,* pp. 1093 1100, 2009.

[17] Conte, D., Foggia, P., Percannella, G., Vento, M.: Counting moving persons in crowded scenes, *Machine Vision Application,* 24, 1029 - 1042, 2013.

[18] D. Conte, P. Foggia, G. Percannella, F. Tufano and M. Vento, Counting moving people in videos by salient points detection, *ICPR,* pp. 1743 - 1746, 2010.

[19] C. Cortes, and V. Vapnik, Support-Vector Networks, *Machine Learning,* 20, 1995.

[20] N. Dalal and B. Triggs, Histograms of oriented gradients for human detection, *CVPR,* 2, 886 893, 2005.

[21] J. W. Davis, Recognizing movement using motion histograms, *MIT Media lab Technical Report #487,* March 1999.

[22] O. Debeir, P. Van Ham, R. Kiss and C. Decaestecker, Tracking of migrating cells under phase-contrast video microscopy with combined mean-shift processes, *IEEE Transactions on Medical Imaging,* 24, 697 - 711, 2005.

[23] P. Felzenszwalb, D.McAllester, and D. Ramanan, A discriminatively trained, multiscale, deformable part model, *CVPR,*pp. 1 8, 2008.

[24] D. J. Fleet and Y. Weiss, *Optical Flow Estimation,* Handbook of Mathematical Models in Computer Vision, Springer, 2006.

[25] K. Fukunaga, *Introduction to Statistical Pattern Recognition,* Academic Press, second edition, 1990.

[26] W. Ge and R. T. Collins, Marked point processes for crowd counting, *CVPR,* pp. 2913 2920, 2009.

[27] R. C. Gonzalez and R. E. Woods, Digital Image processing, *Prentice Hall*, Third Edition, 2008.

[28] S. Harasse, L. Bonnaud, M. Desvignes, People Counting in Transport Vehicles, *Transactions on Engineering, Computing and technology*, 4, 221 - 224, 2005.

[29] B. K. P. Horn and B. G. Schunck, Determining optical flow, *Artif. Intell.*, 17, 185 203, 1981.

[30] C. P. Hou, C.S. Zhang, Y. Wu, F.P. Nie, Multiple view semi-supervised dimensionality reduction, *Pattern Recogn,* 43, 720 - 730, 2010.

[31] C. Hua, Y. Makihara and Y. Yagi, Pedestrian Detection by Combining the Spatio and Temporal Features, *MIRU*, 2010.

[32] K. Jaqaman, D. Loerke, M. Mettlen, H. Kuwata, S. Grinstein, S.L. Schmid and G. Danuser, Robust single-particle tracking inlive-cell time-lapse sequences, *Nature Methods,* 5, 695 - 702, 2008.

[33] P. Jain, R.A. Worthylake and S.K. Alahari, Quantitative Analysis of Random Migration of Cells Using Time-lapse Video Microscopy, *Journal of Visualized Experiments,* 63, e3585, 2012.

[34] J.W. Kim, K.S. Choi, B.D. Choi and S.J. Ko, Real-time vision-based people counting system for the security door, *ITC-CSCC,* pp. 1418 - 1421, 2002.

[35] Y.-S. Kim, G.-G. Lee1, J.-Y. Yoon, J.-J. Kim, and W.-Y. Kim, A method of counting pedestrians in crowded scenes, *International Conf. on Intelligent Computing,* pp. 1117 - 1126, 2008.

[36] N. Krahnstoever, T. Yu, K.A. Patwardhan, D. Gao, Multi-Camera Person Tracking in Crowded Environments, *PETS-Winter Workshop,* 1, 1 - 7, 2008.

[37] K. Li, E.D. Miller, M. Chen, T. Kanade, L.E. Weiss, P.G.Campbell, Cell population tracking and lineage construction with spatiotemporal context, *Medical Image Analysis,* 12, 546 - 556, 2008.

[38] V. Lipetit and T. Fua, Monocular model based 3D tracking of rigid objects: a survey, *Foundations & trends in comp. graph. & vis.,* 1, 1 - 89, 2005.

[39] B. D. Lucas and T. Kanade, An iterative image registration technique with an application to stereo vision, *IJCAI*, 1981.

[40] Z. Ma and A.B.Chan, Crossing the Line: Crowd Counting by Integer Programming with Local Features, *CVPR,* pp. 2539 - 2576, 2013.

[41] N. J. B. McFarlane and C. P. Schofield, Segmentation and tracking of piglets in images, *Machine Vision and Applications*, 8, 187 - 193, 1995.

[42] N. Ray, G. Dong and S.T. Acton, Tracking multiple cells by correspondence resolution in a sequential bayesian framework, *Proceedings of IEEE Conference on Image Processing,* pp.705 - 708, 2005.

[43] V. Salari and I.K. Sethi, Feature point correspondence in thepresence of occlusion, *IEEE Transactions on Pattern Analysisand Machine Intelligence,* 12, 87 - 91, 1990.

[44] I.F. Sbalzarini and P. Koumoutsakos, Feature point tracking andtrajectory analysis for video imaging in cell biology, *Journal of Structural Biology,* 151, 82 - 195, 2005.

[45] M. Sezgin and B. Sankur, Survey over image thresholding techniques and quantitative performance evaluation, *Journal of Electronic Imaging,* 13, 146 - 165, 2004.

[46] I. Smal, K. Draegestein, N. Galjart, W. Niessen, E. Meijering, Particle filtering for multiple object tracking in dynamic fluorescence microscopy images: application to microtubule growth analysis, *IEEE Transactions on Medical Imaging,* 27, 789 - 804, 2008.

[47] A. Smola and B.Scholkopf, A Tutorial on Support Vector Regression, Neuro-COLT Technical Report, NC-TR-98-030, Royal Holloway College, University of London, UK, 1998.

[48] C. Stauffer and W. Grimson, Adaptive background mixture models for real-time tracking, *CVPR,* 2, 246 - 252, 1999.

[49] Y. Mu, S. Yan, Y. Liu, T. Huang and B. Zhou, Discriminative Local Binary Patterns for Human Detection in Personal Album, *CVPR,* pp. 1 - 8, 2008.

[50] S. Mukherjee, B. Saha, I. Jamal, R. Leclerc, N. Ray, A novel framework for automatic passenger counting, *IEEE ICIP*, 2011.

[51] S. Mukherjee and N. Ray, A Novel Framework for Unique People Count from Monocular Videos, Accepted and Received Best PhD Student Award at *VISIGRAPP,* 2014.

[52] S. Mukherjee and N. Ray, Unique People Count From Monocular Videos, Submitted to *ICIP* 2014.

[53] S. Mukherjee, S.Gil and N. Ray, A Novel Framework for Unique People Count from Monocular Videos, Submitted to *The Visual Computer Journal (TVCJ).*

[54] S. Mukherjee, N. Ray and S. Acton, Counting cells from microscopy videos without tracking individual cells, Accepted to be published in *IEEE ISBI,* 2014.

[55] S. Mukherjee and N. Ray, DTV: Detection, Tracking and Validation Framework for Unique People Count, *IJCSNS* Vol 2. No.1, 2014.

[56] T. Ojala, M. Pietikainen and T.Maenpaa, Multiresolution gray scale and rotation invariant texture analysis with local binary patterns, *PAMI*, 2002.

[57] V. Rabaud and S. J. Belongie, Counting crowded moving objects, *CVPR,*, pp. 705 711, 2006.

[58] C. E. Rasmussen and C. K. I. Williams, Gaussian Processes for Machine Learning, *Cambridge, MA : MIT Press*, 2006.

[59] B. Tan, J. Zhang and L. Wang, Semi-supervised elastic net for pedestrian counting, *Pattern Recognition*, 44, 2297 - 2304, 2011.

[60] R.Tibshirani, Regression shrinkage and selection via the lasso, *Journal of the Royal Statistical Society*, Series B(Methodological) 58, 267 288, 1996.

[61] P. Viola and M. Jones, Robust Real-time Object Detection, *Second International Workshop on statistical and computational theories of vision-modeling, learning, computing, and sampling*, 2001.

[62] P. Viola, M. Jones, and D. Snow, Detecting pedestrians using patterns of motion and appearance, *Int. J. Computer Vision*, 63, 153 - 161, 2005.

[63] B. Wu and R. Nevatia, Detection of multiple, partially occluded humans in a single image by Bayesian combination of edgelet part detectors, *Proc. IEEE Int. Conf. Comput. Vis.*, 1, 90 97, 2005.

[64] C. Zeng and H. Ma, Robust head-shoulder detection by PCA-based multilevel HOG-LBP detector for people counting, *ICPR*, pp. 2069 - 2072, 2010.

[65] T. Zhao and R. Nevatia, Bayesian human segmentation in crowded situations, *CVPR,* 2, 459 466, 2003.

[66] X. J. Zhu, Semi-supervised learning literature survey, Technical Report 1530. Computer Sciences, University of Wisconsin-Madison, USA, 2005.

[67] V. Lempitsky and A. Zisserman, Learning to count objects in images, *NIPS,* 2010.

[68] X. Zhou and Y. Lu, Efficient mean shift particle filter for sperm cells tracking, *International Conference on Computational Intelligence and Security,* pp. 335 - 339, 2009.

[69] H.Zou and T.Hastie, Regularization and variable selection via the elastic net, *Journal of the Royal Statistical Society*, Series B(Methodological) 67, 301 320, 2005.

[70] `https://webdocs.cs.ualberta.ca/~satarupa/SDF.mpg`

[71] `http://www.svcl.ucsd.edu/projects/motiondytex`

[72] `http://www.iipl.fudan.edu.cn/~zhangjp/Dataset/fd$_$pede$_$dataset$_$intro.htm`

[73] `http://www.gaussianprocess.org/gpml/code/matlab/doc/`

[74] `http://www.mathworks.com/matlabcentral/fileexchange/22756-horn-schunck-optical-flow-method`

[75] `http://www2.ulg.ac.be/telecom/research/vibe/`

[76] `http://http://www.cvg.rdg.ac.uk/PETS2013/a.html#s1`