Robust Pedestrian Tracking using Improved TLD Algorithm

Dissertation submitted in
Partial fulfilment of the requirement
For the award of the degree of

Master of Technology

in

VLSI and Embedded System Design

by

Ritika Verma

University Roll No. 2K14/VLS/17

Under the guidance of

Dr. S. Indu
Associate Professor,
Electronics and Communication Engineering Department, DTU

2014-2016

ELECTRONICS AND COMMUNICATION ENGINEERING DEPARTMENT

DELHI TECHNOLOGICAL UNIVERSITY

DELHI-110042, INDIA

.

Department of Electronics and Communication Engineering

Delhi Technological University

Delhi-110042

www.dce.edu

CERTIFICATE

This is to certify that the dissertation titled **"**Robust Pedestrian Tracking using improved TLD algorithm**"** is a bonafide record of work done by **Ritika Verma , Roll No. 2K14/VLS/17** at **Delhi Technological University, Delhi** for partial fulfilment of the requirements for the degree of Master of Technology in VLSI and Embedded System Design. This project was carried out under my supervision and has not been submitted anywhere else, either in part or full, for the award of any other degree or diploma to the best of my knowledge and belief.

Date: _____

(Dr. S. Indu)
Associate Professor
Department of Electronics and Communication Engineering
Delhi Technological University

ACKNOWLEDGEMENT

This work would have not been possible without the support of many. First, it is my duty to thank God Almighty for making the completion of this research possible. Next, I would like to express my gratitude to my parents and my family for their continued support.

I would like to express my deep sense of respect and gratitude to my project supervisor **Dr S. Indu ,** Associate Professor, Electronics and Communication Engineering Department, DTU for providing the opportunity of carrying out this project and being the guiding force behind this work. I am deeply indebted to her for the support, advice and encouragement she provided without which the project could not have been a success.

I am also grateful to **Prof. Prem R. Chadha**, HOD, Electronics and Communication Engineering Department, DTU for his immense support.

I would also like to acknowledge Delhi Technological University for providing the right academic resources and environment for this work to be carried out.

Last but not the least I would like to express sincere gratitude to my friends and my colleagues for constantly encouraging me during the completion of work.

Ritika Verma
University Roll no: 2K14/VLS/17
M.TECH. (VLSI Design and Embedded System)
Department of Electronics & Communication Engineering
Delhi Technological University
Delhi – 110042

ABSTACT

Object tracking is defined as the estimation of location of an object of interest in image sequence, whose initial position is defined in the first frame. It has applications in the field of surveillance, traffic management, sports event monitoring and most recently in Driverless assistance systems .Most existing digital video surveillance systems rely on human observers for detecting specific activities in a real-time visual scene. However, there are limitations in the human capability to monitor simultaneous events in surveillance displays. Hence, human motion analysis in automated visual surveillance has become one of the most active and attractive research topics in the area of computer vision and pattern recognition.

Earlier methods of object tracking used either tracking or detection, neither of which were independently sufficient for tracking under complex situation .TLD suggested by kalal et.al. [1] is an award winning technique in which ,tracking and detection are integrated along with a new learning component called P-N learning .These 3 components forms a strong feedback loop. Comparison of TLD with earlier tracking methods shows that TLD has better performance in many aspects of difficulty such as - long lengths of video, occlusion, zoom and background clutter.

In this thesis, implementation of the TLD algorithm is described in detail. It is evaluated specifically from pedestrian tracking prospective. We extended TLD to track multiple objects. Harris features [2] are added to benefit the algorithm by providing robustness to out-of plane rotation of the object in image sequence. Automatic initialization is also accomplished using Histogram of oriented gradients [3].

Experiments are conducted on complex datasets and results are compared. Implementation results shows that the final result is robust to occlusion and worked on frame rates comparable to real life scenarios. The final implementation also provides the trajectory of each pedestrian which can be used for crowd flux analysis.

LIST OF FIGURES

LIST OF TABLES

INDEX

CHAPTER 1

INTRODUCTION

1.1 MOTIVATION

Tracking is performed in the context of higher-level applications that require the location and/or shape of the object in every frame. Object tracking is an active area of research and many tracking algorithms have been suggested in the past. A detailed survey of tracking algorithms in provided by Yilmaz et. al. [4].

Applications of object tracking are:

1. Pedetrian tracking and surveillance

Tracking pedestrians in real time can be very time consuming especially in large crowds there will be a number of pedestrians and tracking each one of them individually will be very tiring.

2. Person specific identification

Another application of smart surveillance system is that it can help police to catch suspects. This requires a database of biometric features of the suspect .visual surveillance system can be placed at location where suspects usually appears e.g., subway stations, casinos, etc. The systems automatically recognize whether or not the people in view are suspects.

3. Crowd flux statistics and congestion analysis

Human detection techniques combined with visual surveillance can automatically compute the flux of people at important public areas such as travel sites .The data obtained from tracker can then provide congestion analysis to assist in the management of people based on the crowd behaviour.

4. Traffic management

Visual surveillance systems can be used for analysis of traffic flow, which can provide the status of road congestion .In this way visual surveillance systems can monitor expressways and junctions of the road network. Interactive surveillance using multiple

cameras can help the traffic police discover, track, and catch vehicles involved in traffic offences.

Other applications include analysis of animal behaviour in forests where human intervention is difficult and sports–event monitoring .Object tracking is finding rising importance in Driverless assistance systems.

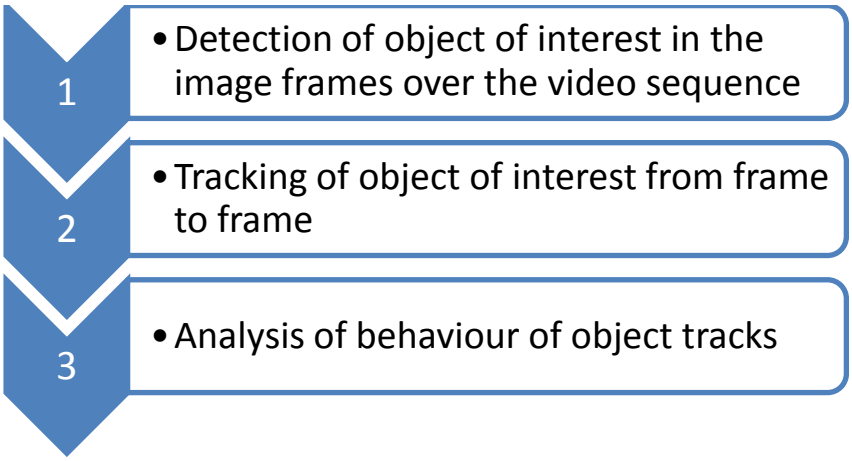


Fig. 1.1 Steps involved in object tracking

Many issues faced in tracking such as scale change, illumination change, background clutter and occlusion. Occlusion is said to occur when the object of interest or its key attributes used for tracking, is not available for camera sensor while the object is still present in the scene. When one part of the object occludes another part, this is known as self occlusion Inter-object occlusion occurs when two objects being tracked occlude each other. Similarly, occlusion by the background occurs when a structure in the background occludes the tracked objects. Another important factor to be considered is real time performance.

1.2 THESIS OBJECTIVE

The main aim of this thesis is to achieve robust multi-pedestrian tracking which overcome issues such as occlusion, illumination change, scale change and background clutter and also work in real time .It should also be able to trace the trajectory of pedestrians and should be automatically initialised .

1.3 THESIS ORGANISATION

The next chapter discusses object tracking in detail. The discussion includes the methods for general representations of object and the requirements of feature selection. A brief survey of existing methods of multi-object tracking is also presented.

Chapter 3 discusses tracking learning detection framework. The operation of 3 subtasks – tracking, learning and detection have been discussed in detailed. A section on integration is added to describe how these 3 components interact to produce final result. The process flow to describe the implementation of algorithm is also added. This chapter ends with analysis of TLD and its limitations, along with proposed methodology to overcome these limitations.
The method by which Harris corner features are detected in the patch under consideration been studied in chapter 4, and the mathematical formulation for the same has been provided. In chapter 5, a detailed discussion has been done on how Histogram of oriented gradients is used to detect pedestrians in the image. Chapter 6 shows the experimental procedure and results of this implementation. The conclusions and future scope are discussed in chapter 7.

CHAPTER 2

OBJECT TRACKING

The object of interest depends on the specific application .For traffic surveillance the object of interest will be vehicles on road, for pedestrian tracking –people walking in road and for animal behaviour –animals in forest.Object can be represented by shape and appearance.

2.1 SHAPE REPRESENTATION

The various object shape representation commonly employed for tracking are described as –

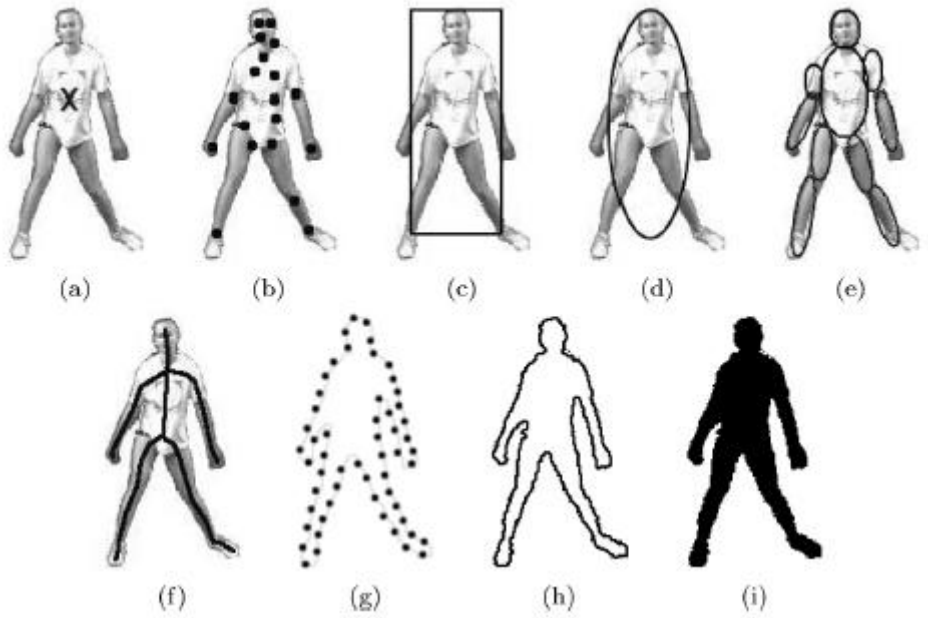


Fig. 2.1 Object representations. (a) Centroid, (b) Multiple points, (c) Rectangular patch, (d) Elliptical patch, (e) Part-based multiple patches, (f) Object skeleton, (g) Control points on object contour, , (h) Complete object contour, (i) Object silhouette.[4]

(A)Points

Object can be represented by centroid as shown in figure 1(a) or by a set of points as shown in figure 1(b). For tracking objects, which occupy small portion of the image, point representation is appropriate.

(B)Primitive geometric shapes

For objects whose shape closely resembles rectangles or ellipses, primitive geometric shape representations can be used to encode their appearance as shown in figure 1 (c) and figure 1 (d).

(C) Object silhouette and contour

Contour expresses the outline of an image as shown in figure 1(g), (h). The region inside the contour is called the silhouette of the object (figure 1(i)). Silhouette and contour representations are suitable for tracking complex non-rigid shapes because it allows maximum freedom to change in object appearance. For tracking objects with complex shapes, for example, pedestrian tracking and biomedical applications it is found highly useful.

(D) Articulated shape models

The basic idea behind this model is to consider object as group of rigid parts held together by joints .For example ,human body is composed of small parts like legs, head, hands and feet represented by cylinders or ellipses as shown in figure 1(e). Kinematic motion models are used to describe the relationship between these parts.

(E) Skeletal models

It is used to represent shape for both articulated and rigid objects as shown figure 1(f).It is implemented by employing medial axis transform method [5] on silhouette of the object.

2.2 APPEARANCE REPRESENTATION

Appearance representation implies a consistency of certain property to transfer one frame to the next for example brightness consistency. Appearance representations in the context of object tracking are:

(A) Probability densities of object appearance

It can be

1) Parametric, such as Gaussian [6] and a mixture of Gaussians [7].
2) Non-parametric, such as Parzen windows [8] and histograms [9].

The advantage of using histogram representation is that it removes spatial ordering constraint giving room for flexibility of the target while moving. Image regions for calculating the probability densities of object appearance features like color and texture are specified by the shape models.

(B) Templates

Template carries both spatial and appearance information and can be formed using simple geometric shapes or silhouettes. Template method has limitations as it considers only a single appears of the object and fails if appearance of the object changes.

(C) Active appearance models

Active appearance models are a group of highly flexible deformable models that are generated by simultaneously using the object shape and appearance. Here, the object shape is defined as a set of parameters to characterize the identity, pose, expression, lighting etc. Active appearance models require a training phase by applying principal component analysis to labeled images, where both the shape and its associated appearance is learned from a set of samples.

(D) Multi-view appearance models

These models encode different views of an object. One approach to represent the different object views is to generate a subspace from the given views.

1) Subspace approaches ,i.e. by developing a subspace from given views such as Principal Component Analysis (PCA) and Independent Component Analysis (ICA) method.

2) By training a set of classifiers to learn the different views of an object, for example, the support vector machines or Bayesian networks.

One limitation of multi-view appearance models is that the appearances in all views are required ahead of time for learning.

2.3 FEATURE SELECTION FOR TRACKING

Feature recognition is first step of object recognition. A local feature is an image pattern which differs from its immediate neighborhood. Good features have following properties [10].

(A)Repeatability

Irrespective of the viewing condition, features should be detected on the object part visible in both images.

(B) Distinctiveness

An interest point can be easily discriminated against the background. It should also possess a global uniqueness, in order to improve the discrimination of repetitive patterns.

(C) Quantity

The detected features should be sufficiently large in number so that reasonable number of features is detected even on small objects.

(D) Accuracy

Points should be robust to errors in feature detection process.

(E) Efficiency

Preferably, the detection of features in a new image should allow for real time applications.

Repeatability is the most important property of all and can be achieved in two different ways:

(A) Invariance

The determination of features should be independent of the geometrical distortions.

(B) Robustness

The selection of interest points should be robust to noise.

The desired feature can be selected based on the application as each one has its own strength and weakness'.

(A) Color

Color of an object is influenced primarily by two physical factors

1) the spectral power distribution of the illuminant and
2) the surface reflectance property of the object.

In image processing, the RGB (red, green, blue) and HSV (Hue, saturation and value) color space are usually used.

(B) Edges

Image intensities change at object boundaries. Edge detection helps in identify boundaries. As compared to color features they are less sensitive to illumination changes. Algorithms that track the boundary of the objects usually use edges as the representative feature. Canny Edge detector is the most popular edge detection approach because of its simplicity and accuracy.

(C)Optical Flow

Optical flow defines the translation of each pixel in a region from one frame to the next in terms of dense field of displacement vectors. Here, brightness constraint is considered which assumes brightness constancy of corresponding pixels in consecutive frames. Optical flow is commonly used as a feature in motion-based segmentation and tracking applications and is also used in the TLD algorithm. Techniques for computing dense optical flow include methods by Lucas and Kanade [11] and Horn and Schunck [12].

(D)Texture

Texture is a measure of the intensity variation of a surface .It quantifies properties such as smoothness and regularity. One such feature is Local Binary Pattern for texture.

(E)Gradient features

Such as SIFT descriptor [13], SURF descriptor [14], HOG descriptor [3] etc. The selection of these features is based on application such as HOG feature is commonly used for detection of human .It is described in chapter 5.

Various combinations of existing features are also used for improved performance.

2.4 SURVEY OF EXISTING METHODS OF MULTIOBJECT TRACKING

Pedestrian tracking aims for precise location estimate of target humans in real life scanarios. It requires maintaining their identities during the entire image sequence which involves 2 steps –detection of pedestrians and their association between frames. Many models are suggested for pedestrian detection.

On such method is Markov chain Monte Carlo data association method, using joint probabilistic data association (JPDA) for multi-target tracking [15]. In another method called Hybridboosted multi-target tracker, target trajectories are acquired by combining detection outputs into tracklets. Ranking is done before association by comparison with other alternatives and wrong associations are removed by classification part [16].Tracking by-detection algorithm for multi-person tracking is suggested in [17] .Here a particle filter is also added to take past information into consideration. Other method using linear programming techniques is proposed in [18].

CHAPTER 3
TRACKING LEARNING DETECTION FRAMEWORK

In this chapter, the TLD framework and its different components are described in detail.

There are 3 components in TLD framework- Tracking, Learning and Detection which work independent of each other. By separating tracking and detection component TLD outperforms earlier methods such as tracking-by-detection [19].

Tracker follows the object from frame to frame whereas the detector considers each frame individually and runs a sliding window on the frame to find the location of the object. Optical flow is for feature correspondence during tracking. The detector uses a cascade approach which leads to considerable reduction in computing time. The task of learning is to estimate and update error in the detector to avoid these errors in future. Error is estimated by a pair of experts, using the P-N learning method [20].

Tracking and detection are 2 separate source of information which provides the bounding box of the object. They cannot solve the real time tracking problem individually but can potentially benefit from each other if used simultaneously. Tracker helps the detector by providing data for training classifier. The relationship between the 3 components is shown in figure 3.1.

Fig. 3.1 Block diagram of TLD framework [1]

TLD works even in the case of occlusion because it keeps on adding new appearances. The source of new appearances is the trajectory path of the tracker .Detector is trained with the appearances found on the trajectory. Now if the object goes under occlusion it will still be detected, when it will come out of occlusion because its appearance was stored in the memory. Without storage of templates in memory, the tracking will be lost to occlusion.

3.1 TRACKING

Tracking is analysis of video sequence for estimation of target's location in every frame over the sequence .Tracking is based on median flow trackers [21] and estimates motion between consecutive frames.

The first step is selection of a fixed number of point numbers of points within the bounding box. These points are selected as a grid of 10X10 points uniformly distributed within the bounding box as shown in figure 3.2.

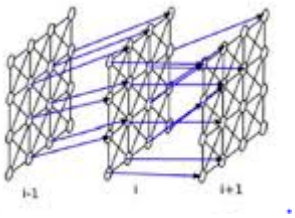


Fig. 3.2 Uniform grid of points tracking

Then the position of these points in the next frame is estimated by Lucas-kanade method [11] .Using these estimated position new bounding box is predicted by analysing the displacement of points in the consecutive frame.

3.1.1 LUCAS-KANDE METHOD

The Lucas-Kanade algorithm technique provides an estimate of the movement of interest points in the consecutive images of video sequence, by using intensity gradients of the image in that neighbourhood It does not scan the second image trying to find a match for the given interest point rather it estimates the direction in which an object has moved by using constraint of local change in intensity.

There are 2 assumptions in this method

(A)Small motion

The two images are separated by a small time increment Δt , points do not move very far,

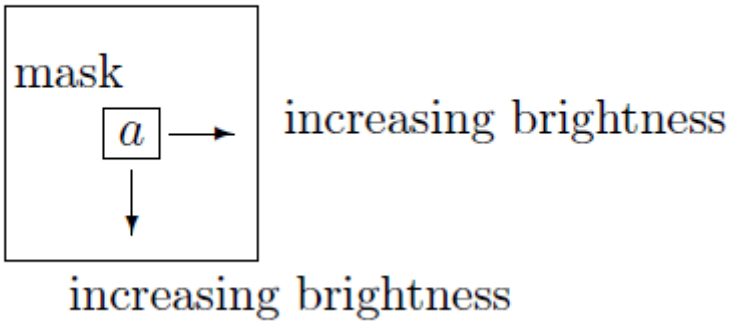
(B)Intensity levels change smoothly.

The need of these two conditions is explained in the next section.

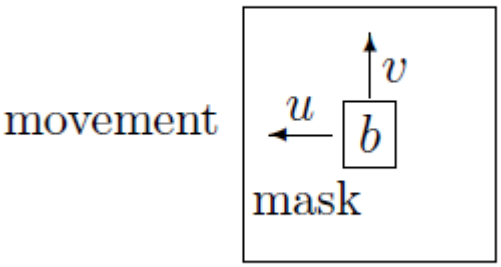
The main task of this algorithm is to associate a movement vector (u,v) to the point being tracked .

3.1.2. TECHNIQUE

To explain this technique [22] a pixel in the image is watched through a square hole which is of intensity ‘a’ initially.



The intensity of the pixel has increased to b in the next frame by moving the hole to left in horizontal direction and y in vertical direction.



$I_x(x, y)$ is the increase in brightness per pixel at pixel (x, y) in the x-direction, and $I_y(x, y)$, increase in brightness per pixel (x, y) in the y direction.After a movement of ‘ u’ pixels in the x direction and ‘v’ pixels in the y direction,

Total increase in brightness = $I_x(x, y).u+ I_y(x, y).v$

Local difference in intensity (b - a) is $I_t(x; y)$.

This implies

$$I_x(x, y).u+ I_y(x, y).v= - I_t(x, y) \quad ..eqn(1)$$

which is fundamental equation of Lucas –Kanade tracker.

3.1.3. NEIGHBOURHOODS

Neighbourhood pixels around pixel (x, y) are useful during matching to provide a structure, since a simple pixel does not provide enough information for matching. For one pixel we have two unknowns (u and v) and one equation (eqn. 1) and we need a neighbourhood in order to get more equations. Mathematically it helps to obtain 9 linear equations by considering 3X3 pixels around a given pixel

$I_x(x+\Delta x, y+\Delta y).u + I_y(x+\Delta x, y+\Delta y).v = -I_t(x+\Delta x, y+\Delta y)$

for $\Delta x = -1; 0; 1$ and $\Delta y = -1; 0; 1$

The linear equations can be represented in a compact form as :

$$S\begin{pmatrix} u \\ v \end{pmatrix} = \vec{t}$$

where S is a 9 X 2 matrix containing the rows $I_x(x+\Delta x, y+\Delta y)$, $I_y(x+\Delta x, y+\Delta y)$ and is a vector containing the 9 terms $-I_t(x+\Delta x, y+\Delta y)$.

The next step is to find the Least Squares solution by multiplying the equation by S^T

$$S^T S\begin{pmatrix} u \\ v \end{pmatrix} = S^T\vec{t}$$

and inverting S^TS, so that

$$\begin{pmatrix} u \\ v \end{pmatrix} = (S^TS)^{-1} S^T\vec{t}$$

The solution given above is possible if S^TS is invertible.

To check for inevitability of S^TS ,we look at its Eigen values by writing it in the form

$$S^TS = U\begin{pmatrix} \lambda 1 & 0 \\ 0 & \lambda 2 \end{pmatrix}U^T$$

where U is a 2X2 unitary matrix.

If $\lambda 1$ or $\lambda 2$, or both,are zero then S^TS is not invertible. By analysing the invertibility of the matrix S^TS through the eigen values of this matrix, Lucas-Kanade algorithm eliminates regions without structure.

3.1.4 CONCLUSION

The algorithm does not use colour information of the image. It fails when there is no structure, that is, gradients are negligible. The result of the algorithm is a set of optical flow vectors distributed over the image which describe the movement of objects in the scene.

To remove erroneous points in Lucas –kanade method , 2 error measures are used

1. Forward-backward error [21].
2. Normalised cross correlation.

3.1.5 FORWARD BACKWARD ERROR

It is based on the assumption that correct tracking does not depend on direction [21]. Here, first a tracker produces a trajectory by tracking point in forward direction, which is of a particular length .Then a validation trajectory is obtained by backward tracking from the point location in the last frame. The two trajectories are compared and forward trajectory is rejected if it is not similar to validation trajectory. This is illustrated in figure 3.3 .In case of point 1, trajectories in the forward and backward direction are similar, so the tracking is correct. But in case 2 the trajectories are different in different direction, which implies incorrect tracking. This can be explained as follows:

Point 2 in frame (b) is in the front of the car and it is on the rear of the car in frame (b) ,since they looked similar tracker predicted the location at rear of car 2 in (b) but this incorrect tracking which is rejected with the help of forward –backward consistency.

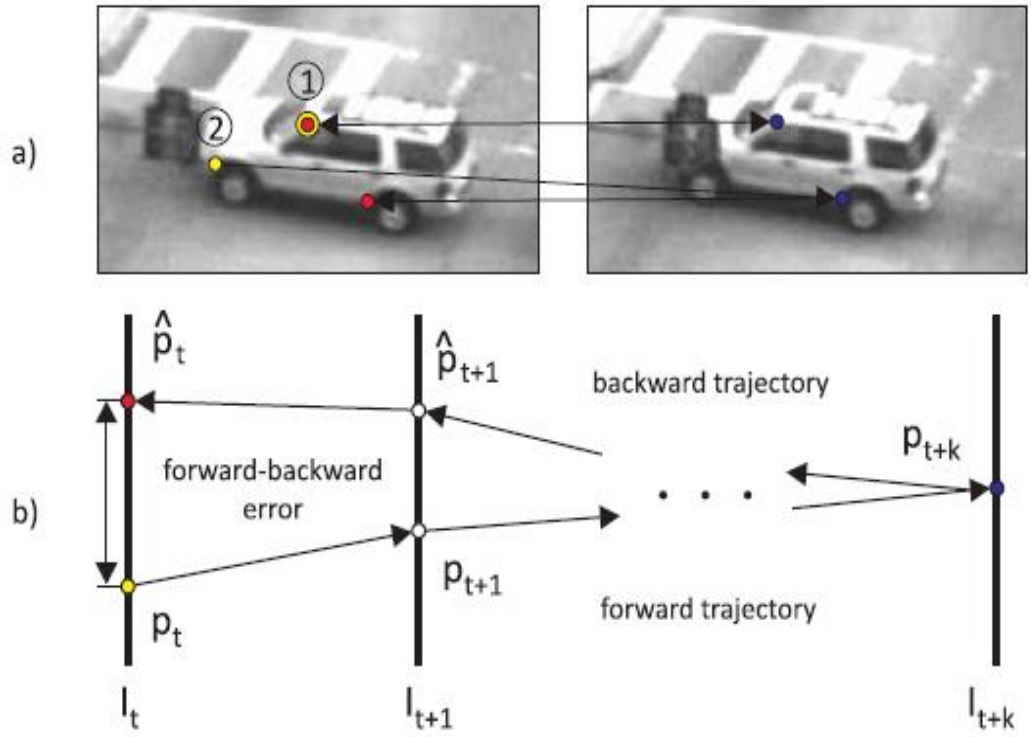


Fig. 3.3 Estimation of error using forward –backward consistency[9]

3.1.6 NORMALISED CROSS CORRELATION

NCC is obtained by subtracting mean and dividing the standard deviation.the cross correlation of a template t(x,y) with a subimage f(x,y) with n pixels is

$$\frac{1}{n}\sum_{x,y}\frac{(f(x,y)-\overrightarrow{f})(t(x,y)-\vec{t})}{\sigma_f\sigma_t}$$

Where f bar is average of f and σ_f is standard deviation of f , t bar is average of t and σ_t is standard deviation of t.

Bounding box is returned if forward backward error is less than median forward backward error (med_{FB}) and similarity measure is larger than median normalised cross correlation (med_{NCC}).These 2 measures filter out the points with errors and increase the reliability of tracking.

To overcome failure due to fast motion and fast occlusion of the target following strategy is used: Let d_i be the displacement of a single point in the median flow tracker .Let d_m be the median displacement .$|d_i-d_m|$ is calculated for all the points and if the median of $|d_i-d_m|>10$ pixels, failure is detected .In case of failure bounding box is not returned.

Problem with trackers is that they accumulate error and drift after some time but in TLD, tracking helps by makes the object model generative rather than static as new appearances on the trackers trajectory are added at each step.

3.2 LEARNING

P-N learning method is contributed by Kalal et .al. in paper [20].In this method learning is done online and it reduces the requirement of preliminary training in offline mode. Tracking and detection are independent processes. They interact and exchange information through learning to benefit each other.

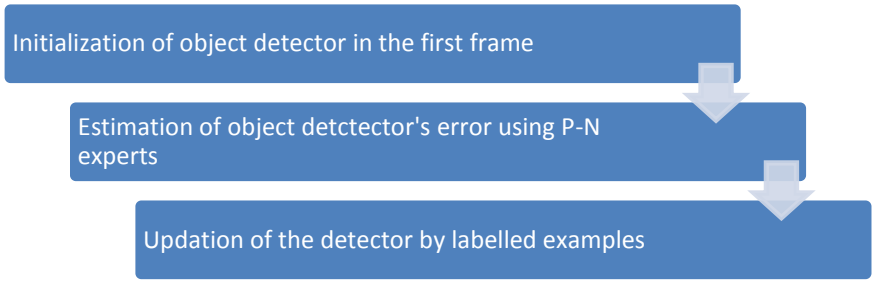


Fig.3.4 Process flow for learning

Learning component initializes the object detector in the first frame and then the training set of the detector is constantly updated and improved by using feedback from these experts, to avoid these errors in future.In this way, learning enables the detector to:

1. Generalise more appearances of the object and

2. Discriminate against the background.

The process flow of P-N learning algorithm is explained in the following steps:

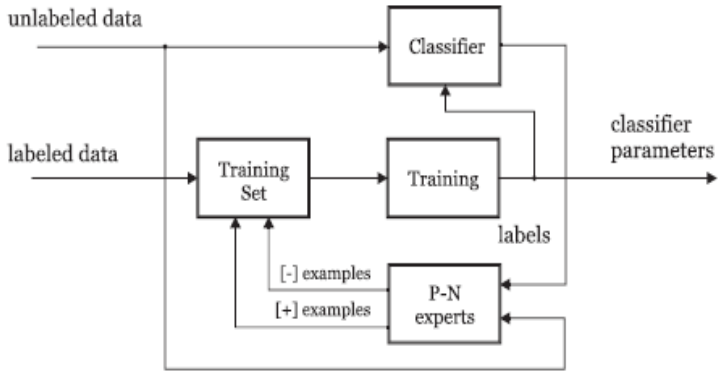


Fig. 3.5 Block diagram of P-N learning [1]

1. L= {(x,y)} called a labelled set is generated by assigning a set of labels Y to set of example X called an unlabeled set.
2. The main task is to learn a classifier f : X ->Y from labelled set L_l and bootstrap its performance by the unlabeled set X_u.
3. A family of classifiers F parameterized by θ is considered which is subject to implementation and is considered fixed in training Classifier f is a function from F .Training corresponds to estimation of the parameters θ.
4. The training process is initialized by inserting the labelled set L to the training set. Semi-supervised learning is used to train the classifier and estimate the estimates the parameter θ.
5. The learning process works iteratively. Detector's results are analysed by P and N experts. P and N expects are introduced for self evaluation and correctness of the algorithm.They analyse which estimate examples are classified incorrectly. These examples are added with changed labels (due to experts) to the training set. Each iteration k finishes by retraining the classifier, i.e., estimation of θ^k. The process repeats itself by convergence or other stopping criterion.

6. For the estimation of the classifier errors the key idea is to separate the estimation of false positives from the estimation of false negatives.
7. P-expert keeps a track of examples classified as negative, estimates false negatives, and adds them with positive label to the training set. In iteration k, P-expert outputs $n^{+}(k)$ positive examples.
8. Examples classified as positive are analysed by N-experts .It estimates false positives, , and adds to the training set them with negative label. In iteration k, the N-expert outputs $n^{-}(k)$ negative examples.
9. The P-expert increases the classifier's generality. The N-expert increases the classifier's discriminality.
10. Labelling is plausible since the object appears at one location in each frame .The detected locations build up a trajectory in time. In other words, the labels of the patches are dependent. We refer to such a property as structure. P-N experts mainly exploit the structure in data to identify the detector errors.
11. P-expert assumes that the object moves along a trajectory and exploits the temporal structure in the video. If the detector labelled the current location as negative (i.e., made false negative error), the P-expert generates a positive example.
12. The N-expert analyzes all responses of the detector and the response produced by the tracker and selects the one that is the most confident. It is called the reference patch.
13. Patches that are not overlapping with the maximally confident patch are labelled as negative. The maximally confident patch reinitializes the location of the tracker.
14. False negatives retrieved by the P-expert are labelled positive and their addition to the set increases detector's generality (it recognizes more appearances of the object), while false positives are labelled as negative by N-expert and increase detectors ability to discriminate against everything that is not the target object.

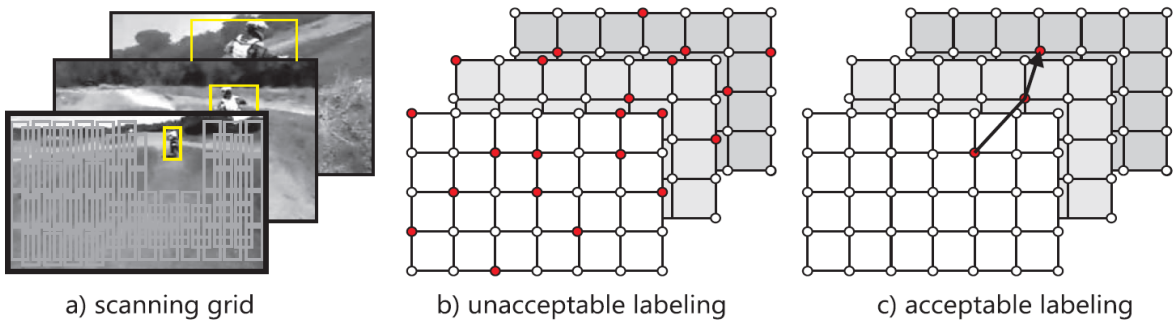


Fig. 3.6 Using structure for positive and negative labelling [1]

3.3 DETECTION

Detection slides a window throughout the image and decides whether the underlying object is the desired target or not. This produces thousands of sub-patches to be evaluated. To achieve real time performance cascaded approach is used.

NN classifier [23] can be used to measure the similarity with the template with each bounding box for making the decision, but it will be time consuming as it involves computation similarity for each patch .So patches are passed through 2 stages – 1.variance filter and 2.ensemble classifier which reject most of the non object patches .Then the remaining patches are passed through NN classifier stage. So a cascade approach is used as shown in figure 3.7

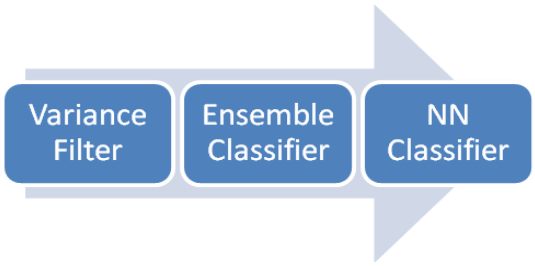


Fig. 3.7 Detection cascade

3.3.1 VARIANCE FILTER

Variance is a measure of similarity .Uniform surfaces such as sky, ground and background regions such as streets, involves low value of variance as shown in figure 3.8. Small value of variance means closeness to the mean value and therefore variance is measure of similarity. Gray value variance of a patch is calculated as $E(p^2)$-$E^2(p)$ where E(p) is the calculated using integral images [24]. It is described in detail in [25].If the Gray value variance is less than 50 percent of the variance of the target patch ,it is rejected.

It rejects a large number of non-object patches, which are not passed to the next stages ,and helps in reducing computation. So effectively background patches are removed in the first stage of the cascade.

Fig. 3.7 Uniform surfaces-ground and sky

3.3.2 ENSEMBLE CLASSIFIER

Patches that are not rejected by the variance filter are passed to ensemble classifier. Here ensemble classification methods called as random fern classification [26] is employed.

The first step is convolution with Gaussian kernel with a standard deviation of 3 pixel .It is done to increase robustness to image noise. The next step is pixel comparisons which yields a result of 0 or 1.Then these results are concatenated into an array x ,which indexes to an array of posteriors Pi(y|x) where y belong to set {0,1}.

Posterior probabilities are the conditional probability obtained after taking large number of relevant evidences into consideration. Here, each base classifier has a distribution of posterior probabilities obtained by pixel comparisons, which are done at random.The Posterior probabilities of individual base classifiers are then averaged. The ensemble classifies it as a patch if the average is greater than 50%.

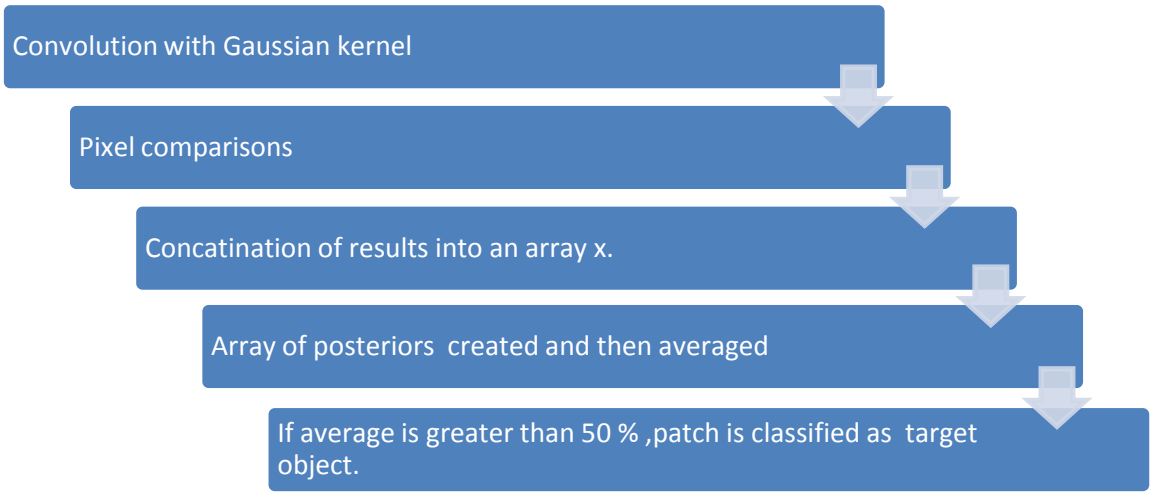


Fig. 3.8 Process flow for ensemble classifier stage

Each base classifier is independent of each other because they perform different pixel comparisons. This is enforced by the following method:

1. All possible vertical and horizontal pixel comparisons are generated
2. Then we permute the comparisons and divided them into base classifiers

As a result each classifier obtains a different set of features .The union of all the features covers the entire patch.

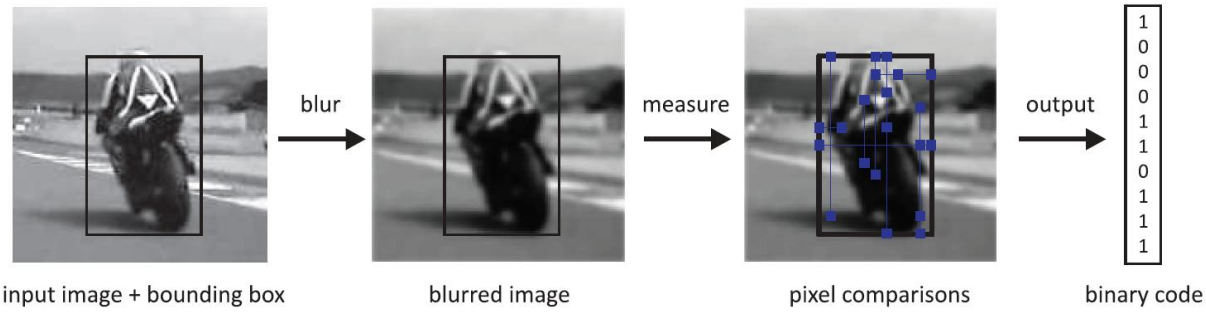


Fig. 3.9 Pixel comparisons to generate binary codes [1]

Random tree classification tree is fast and robust while remain reasonably easy to train [24]. Comparisons are based on intensity values of the pixels makes this method robust to constant brightness variations.

3.3.3 NN CLASSIFIER

After passing through variance filter and ensemble classifier, we are left with reasonable number of patches (approx 50 based on the size of initial bounding box) which are needed to be decided now. NN classifier [23] can be adopted on this small number of remaining patches.

Here, comparison is done pixel by pixel. Similarity between two patches Pi and Pj is calculated by using the formula

$$S(Pi, Pj)=0.5(NCC(Pi, Pj)+1) ,$$

Where NCC is normalised cross correlation.

Patch is classified as an object if similarity is greater than parameter θ. θ is set to 0.6 in experiment .It can be set between 0.5-0.7; however similar performance is obtained [2].

The output of this stage represents the output of object detector and its confidence is obtained by using similarity measure.

3.4 INTERGRATION

The next step is fusion of result of recursive tracker and object detector into a single result. At the end of each iteration both tracker and detector both provide bounding box.The detector identifies the location of the object using previously identified templates, whereas the tracker extracts the location of the object using motion of the object from frame to frame. Maximally confident detector patch or the tracked bounding box becomes the new output after comparison.

If both tracker and detector does not provide bounding box, then it is considered than object is not visible in the current frame. The relationship between the 3 components in shown in figure 3.10.

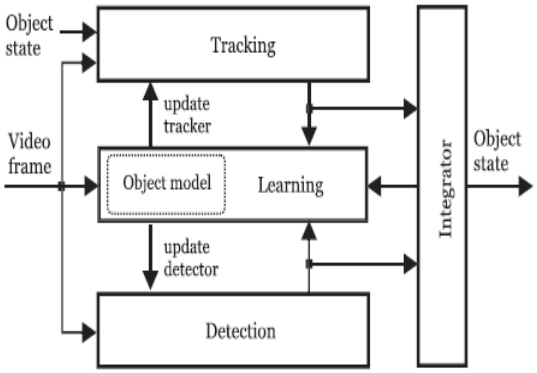


Fig. 3.10 Integration process [1]

Algorithm for integration: Let T represent tracking result and D represent detector result.

If T~=0

If |D| ==1 && conf (D)>conf (T)

Result =D

Else

Result =T

Else if |D|==1

Result =D

For all other cases object is considered invisible.

3.5 PROCESS FLOW OF TLD ALGORITHM

A set of grayscale patches are generated, when the initial bounding box is selected to describe the object and its surroundings. In TLD, target is represented by bonding box. Bounding boxes is an attractive option because of low model complexity. Then they are resized to a normalised resolution (15X15 pixels) for processing.

Object model –object s represented by a data structure having combination of $\{p_1+, p_2+, p_3+, p_4+..\}$positive patch and negative patch$\{p_1-, p_2-, p_3-, p_4-..\}$ .positive patches represents object and negative patches represents background.

For tracking only the bonding box from previous frame is required. Tracking is done by calculating the optical flow using Lucas-Kanade method [11] .For detection, sliding window approach is used to identify the patch. All possible scale and shifts of patches are generated .Since the number of sub-patches is large ,a detection cascade is used .it has variance filter ,followed by Ensemble classifier and NN classifier .Similarity between two patches Pi and Pj is calculated by using the formula

$$S(Pi, Pj)=0.5(NCC(Pi, Pj)+1) ,$$

NCC is normalised cross correlation.

If the number of templates in NN classifier, which is the last stage of object detection cascade, is greater than threshold, we forget these templates randomly, but usually the number of templates is near 100 which can be easily stored in memory.

The task of learning step is to identify falsely labelled examples and updating the ensemble classifier. The result of tracking step and detection step are compared based on the confidence value of the bounding box provided by each of them. The bounding box with a higher value is selected and tracker is reinitialised if detector has higher confidence value. This process is repeated for every frame in the image sequence.

3.6 LIMITATIONS OF TLD ALGORITHM

Current algorithm by kalal et. al. [1] tracks only a single object which needs to be specified in the first frame by the user. There is no automatic initialisation.

The tracker drifts in case of out of plane rotation and causes errors in results.

Detector is unable to differentiate between objects that exhibit similar appearances. Also it forgets the templates randomly when number of templates exceeds the threshold. Detector is unable to recognise changes in appearance that occur while the tracker is not active as these appearances are not added to the memory.

It does not provide information about the orientation of object of interest and there is no automatic detection of tracking failures.

3.7 PROPOSED METHOLOGY

At the tracking step, feature based tracking is used instead of tracking a grid of 10X10 points. Harris corners [2] are used for optical flow. Harris algorithm detects corners by points that have strong gradients in two orthogonal directions which are tracked from one frame to another by Lucas-kanade method [11]. This elaborated in Chapter 4.

Initialisation is done using Histogram of oriented gradients (HOG) [3] to identify humans which is explained in chapter 5 .This leads to automatic initialisation of the algorithm.

TLD is extended to track multiple objects which are pedestrians in this case .Trajectory is the pedestrians are also obtained .This is explained in chapter 6. Yet, the basic structure of TLD which is a strong feedback loop of tracking, detection and learning ,discussed in chapter 3, remains intact.

CHAPTER4

HARRIS FEATURES

To describe the appearance of objects which are being tracked feature descriptors are used in tracking ,for example, corners ,SIFT(scale invariant feature transform) features [13] and SURF (speeded up robust features) [14].The challenge is to find put which features will be appropriate for tracking in real life scenario which is prone to rotation, scale and illumination change.

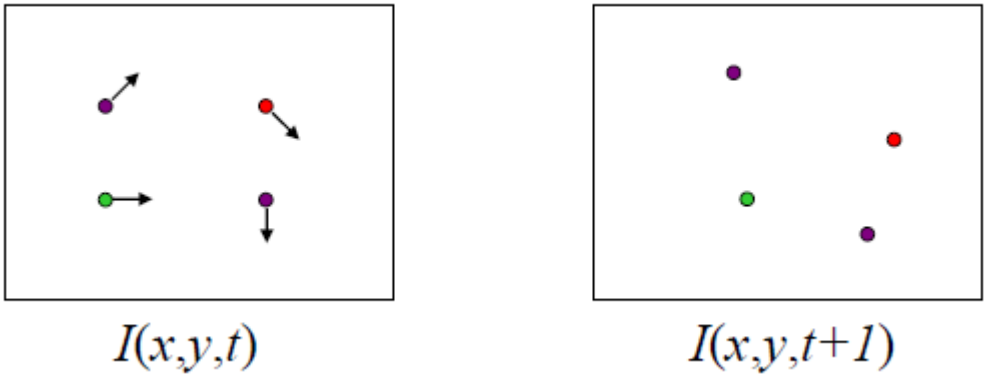


Fig. 4.1 Tracking of feature points [27] .

Combined corner and edge detector was proposed by Chris Harris and Mike Stephans in paper [2].It is based on local auto-correlation function. Earlier edge detection methods like canny edge detection was used .the problem was that small change in edge strength causes large change in edge topology. To solve this problem it was suggested to detect both edge and corner in the image to enable tracking of features.

To detect a corner we need to maximize the variation within a window as described by the Movarec, the precursor of Harris .Moravec considered a local window in the image and determine the average variation of image intensity obtained by shifting the window obtained by a small amount .

The results were as follows:

1. If the image patch is flat ,that is ,constant in terms of intensity, then all shifts will result in only a small change.

2. If edge is present, the shift along the edge will result in small change but shift perpendicular to the edge will result in large change as shown in figure 4.2.

3. If corner is present ,then shift in all the direction will have large change ,so a corner can be detected by finding change produced by shifts is large.

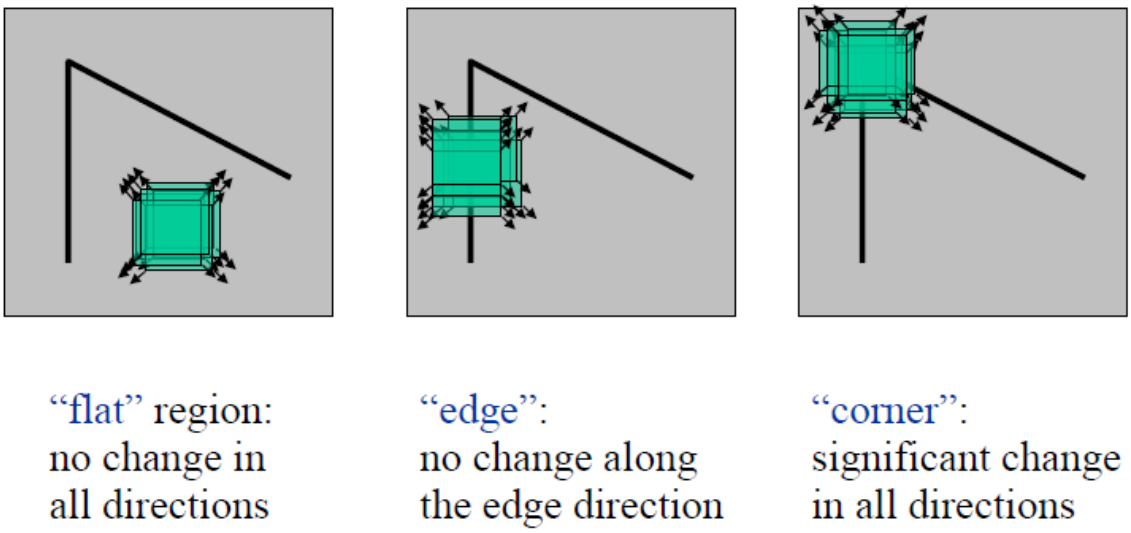


Fig. 4.2 Corner detection [28]

Mathematically it can be represented by:

$$E(u,v)=\sum_{x,y} w(x,y)[I(x+u,y+v)-I(x,y)]^2$$

Where:

w(x,y) is the window weighting function at position (x,y) .It is a binary function ,1 represents within the specified window ,0 represents absence of window.

u :displacement in x direction

v :displacement in y direction

I(x,y)] : intensity at (x,y)

I(x+u,y+v) : intensity at the moved window (x+u,y+v)

The aim is to maximize E(u,v). Applying Taylor expansion and some arithmetic operations, we can get

$$E(u,v) \approx \sum_{x,y} w(x,y)(u^2 I_x^2 + 2uvI_xI_y + v^2 I_y^2)$$

$$= (u, v)\left(\sum_{x,y} \mathrm{w(x,y)} \begin{bmatrix} I_y^x & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}\right)\begin{pmatrix} u \\ v \end{pmatrix}$$

$$= \mathrm{(u,v)M}\begin{pmatrix} u \\ v \end{pmatrix}$$

where (Ix,Iy)is the gradient at (x,y).

Then we calculate the eigenvalues $\lambda 1$ and $\lambda 2$ of matrix M to determine if window corresponds to a corner.

If $\lambda_1$and λ_2 have large positive values, then a corner is found ,as described in point 3 above .

4.1 ALGORITHM [2]

Guassian window used instead of rectangular window and following steps are performed

1.For a image patch ,gradients Ix,Iy are calculated

2. a new matrix 'M' is created using these gradients

$$\mathrm{M}= \begin{bmatrix} I_y^x & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$

M can be represented by $\begin{bmatrix} A & C \\ C & B \end{bmatrix}$

3.Find trace and detminant

$$\mathrm{Tr(M)}=\alpha + \beta = \mathrm{A+B}$$

$$\mathrm{Det(M)} = \alpha\, \beta = \mathrm{AB-C2}$$

4.calculte R response of detector at each pixel

$$\mathrm{R=Det} - k\, \mathrm{Tr}^2$$

5.R is positive for corner region ,negative for edge region ,small for flat region .

To determine corner the algorithm can be simplified as :

The first step is calculating the derivatives Ix and Iy for each pixel in the image, then the distributions of (Ix, Iy) are shown in the figure 4.3 below.

1. For corners, derivates in both Ix and Iy are large.

2.For points on an edge, one derivative has wide distribution while the other is almost all near zeros.

3.For points of a flat area, both derivatives are around zeros.

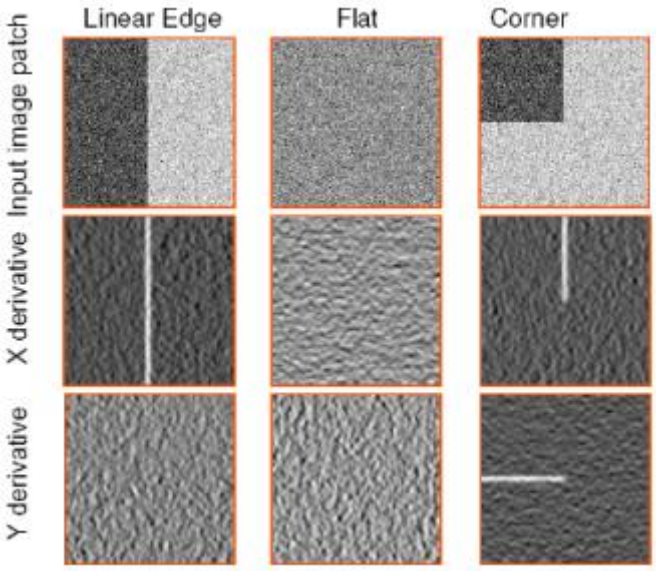


Fig. 4.3 Linear egde, flat and corner points and their derivative in x and y direction [29]

4.2 ROTATION INVARIANCE

Corners are good features to track because even if there is rotation these points will remain the same, from different views and angles .This is contrast with edges whose location will change on changing viewpoint.

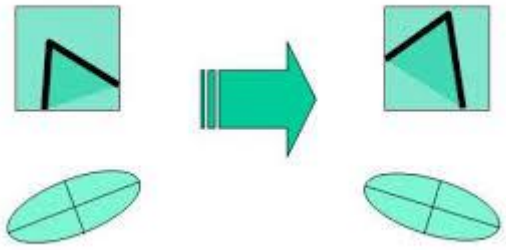


Fig. 4.4 Effect of rotation [30]

As shown in the figure 4.4, ellipse rotates but the corner response, calculated by strong gradient in orthogonal directions, remains the same. This property is called rotation invariance.

4.3FEATURE DETECTION AND MATCHING

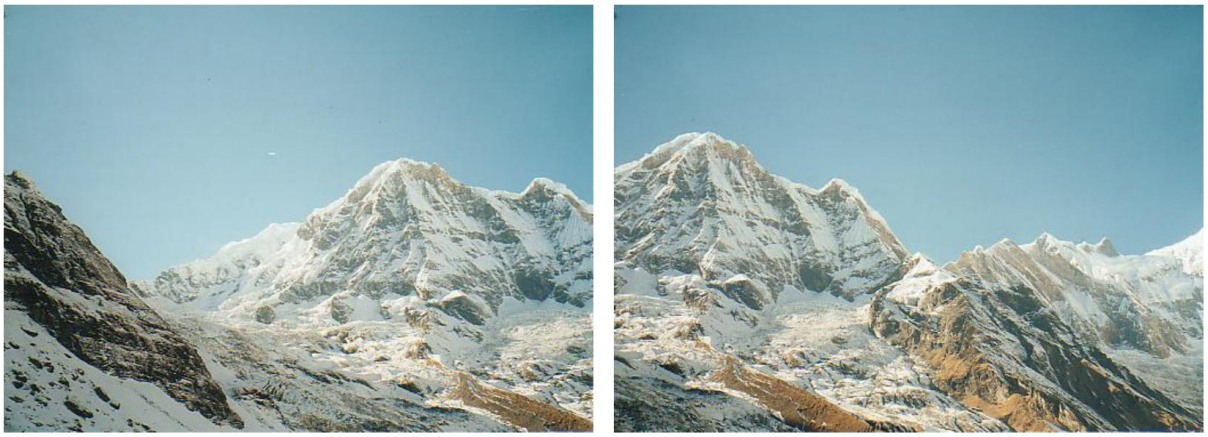


Fig. 4.5 (a) Two consecutive frames in an image

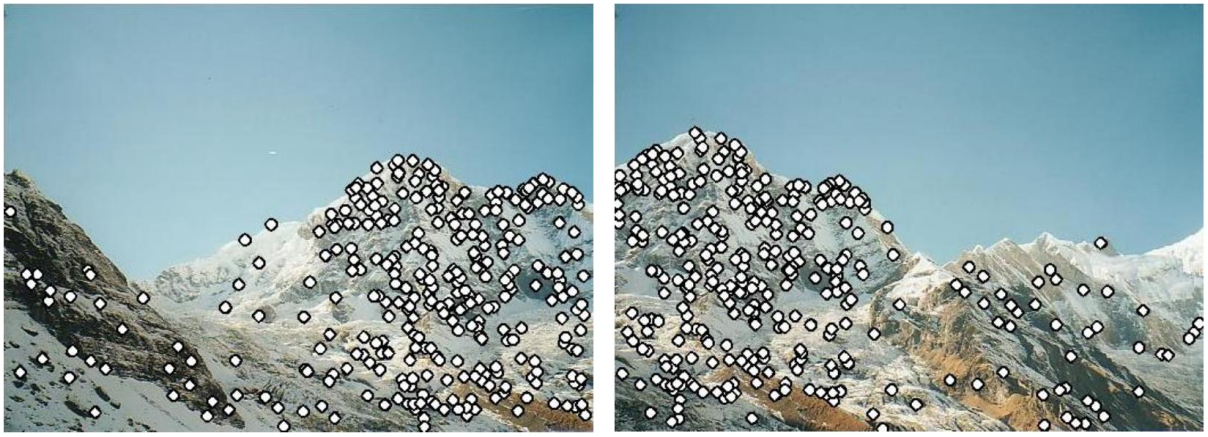


Fig. 4.5 (b) Detection of features

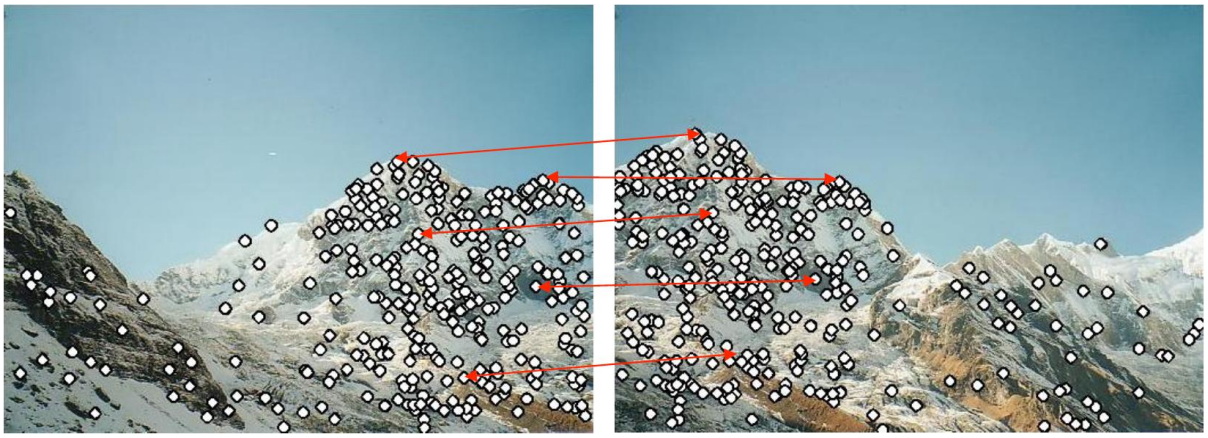


Fig. 4.5 (c) Finding corresponding feature points in images [31]

For feature tracking first the Harris corner points are selected based on the algorithm given in section 4.1 .We also add grid of 5X5 points uniformly surrounding the bounding box to provide neighbourhood required for Lucas–kanade tracking [11] explained in section 3.1.3.

Next step is to apply the Lucas kanade method to track these points from one frame to another .Lucas-kanande method use special intensity information to search for position that yields the best match for corner points between consecutive frames.

The rotation invariance of Harris corner points is discussed section 4.2, which makes them a good candidate for feature tracking and making the algorithm more robust to out of plane rotation.

This method is also computationally beneficial as now the number of points to be tracked is 25 grid pints providing neighbourhood and corner points detected by corner detection algorithm instead of 10X10 grid points, which are used in the original TLD algorithm .

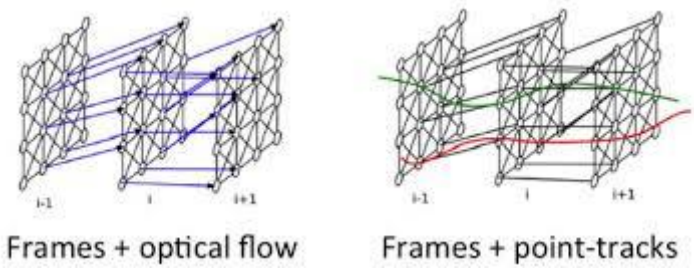


Fig. 4.6 Tracking of Harris points instead of simple grid of points

(a) (b) (c)

Fig. 4.7(a) Tracking using uniform grid of points, **4.7(b)** Harris corners **4.7(c)** Harris corners with neighbourhood

CHAPTER 5

HISTOGRAM OF ORIENTED GRADIENTS

Human detection is a challenging task as because of variable appearances and pose changes .So a robust feature set was need that allowed human form to be uniquely identified ,even in case of cluttered background and illumination changes. Histogram of oriented gradients was introduced by Navneet Dalal and Bill Triggs at the CVPR conference in 2005 [3].They reviewed the existing edge and gradient based methods descriptors and proved that HOG(Histogram of oriented gradients) significantly outperforms existing feature set for pedestrian tracking by conducting experiments on challenging dataset containing over 1800 annotated human images ,which contain pose variations and background variation.

The HOG person detector uses a sliding detection window which is moved around the image. At each position of the detector window, a HOG descriptor is computed for the detection window. The SVM classifier decides whether it is a person or not a person .this is shown on figure 5.1.

Fig. 5.1 Flow chart of HOG detection

5.1 METHOLOGY: This section gives the detailed implementation oh HOG algorithm

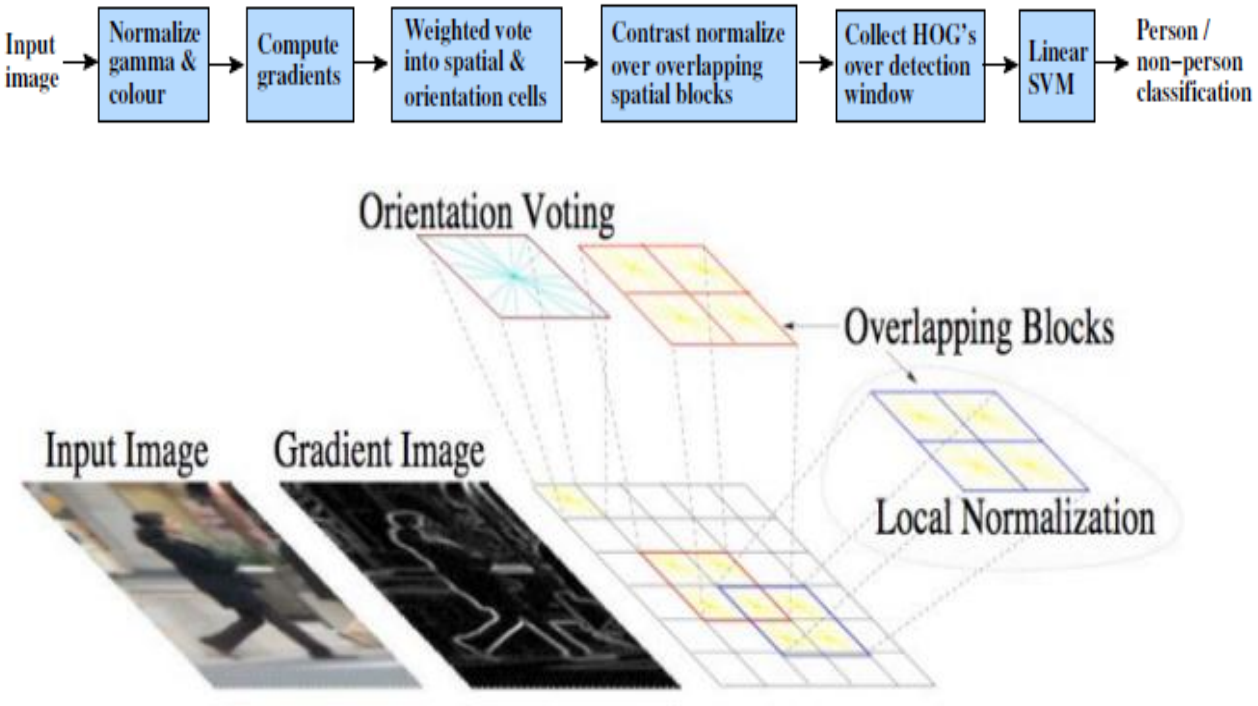


Fig. 5.2 Computation of HOG descriptor in an image[3]

Step1: Input image can be RGB or greyscale image .Dalal and Triggs found that normalization step have modest effect on performance which can be achieved by subsequent descriptor normalization also.

Step2: Several methods of gradient computations were experimented .computation of gradients simple 1 –D mask with σ=0 works the best.

Step 3: Weighted vote into spatial and orientation cell

The HOG person detector uses a detection window that is 64 pixels wide by 128 pixels tall.

Fig. 5.3 Original images used to train the HOG Detector [3]

To compute the HOG descriptor, we operate on 8×8 pixel cells within the detection window.These cells will be organized into overlapping blocks.

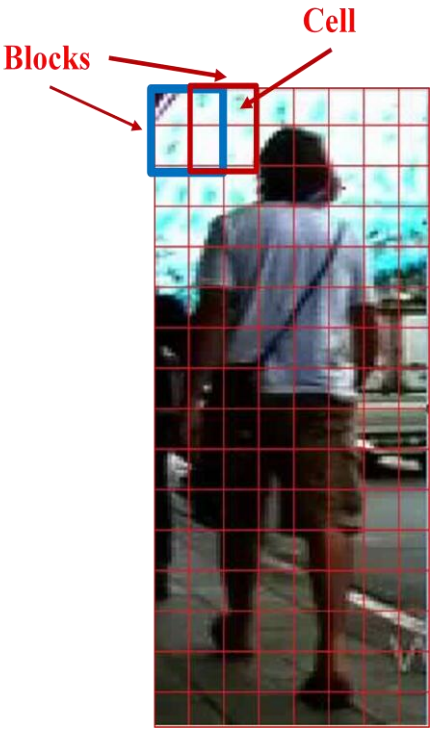


Fig. 5.4 Cells organised into overlapping blocks

We calculate gradient at each pixel to provide 64(8X8 pixel cell) gradient vectors in one cell. Then gradient vectors are put into 9–bin histograms, which ranges from 0 to 180 degrees .This leads to 20 degrees per bin.(180dergees/9 bins =20degrees/bin).

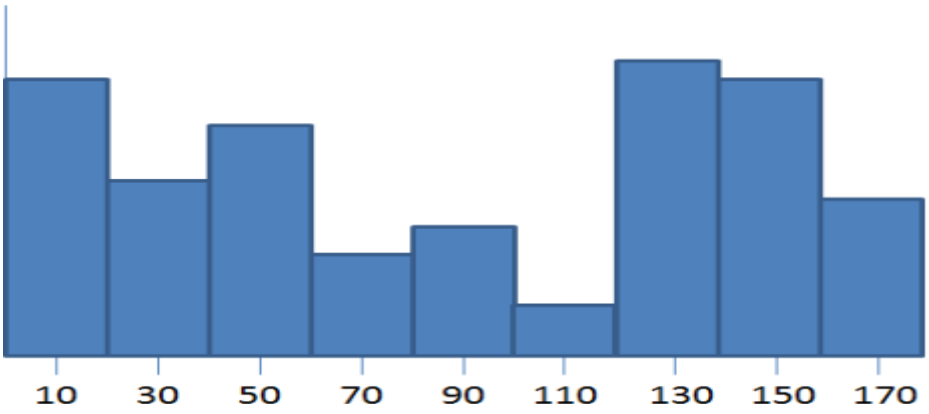


Fig. 5.5 Histogram with 9 bins and 20 degree per bin

For each gradient vector, it's contribution to the histogram is given by the magnitude of the vector , so stronger gradients have a bigger impact on the histogram.

So this leads to reduction of 64 gradient vectors to just 9 values, represented by magnitude of each bin . In this way there is compression of the feature descriptor, which is important for the performance of classifier .the final result is that we can generalise the 8X8 cell by 9 bin values.

It was found by experiment that increasing the number of orientation bins improves performance but only up to 9 bins but beyond this it makes little difference.

5.2 CONTRAST NORMALIZE OVER OVERLAPPING SPATIAL BLOCK

Normalizing refers to dividing a vector by its magnitude. It affects only the magnitude of the vector and not it's orientation .Now the resulting vector will have magnitude 1.

In block normalization histogram of 4 cells within a block are concatenating to produce a vector with 36 components (4 histograms per block X 9 bins in histogram).Then this new vector is normalised .This is contrast to normalizing each histogram individually. Normalisation leads to invariance to brightness and contrast as discussed in section 5.5.

5.3 HOG COLLECTION OVER DETECTION WINDOW

Total number of blocks will be 105 as 64 x 128 pixel detection window will be divided into 7 blocks horizontally and 15 blocks vertically. Each block contains 4 cells with a 9-bin histogram for each cell, for a total of 36 values per block. This brings the final vector size to 7 blocks across x 15 blocks vertically x 4 cells per block x 9-bins per histogram = 3,780 values.so concatenation of histogram produces 1 D matrix of 3780 values .

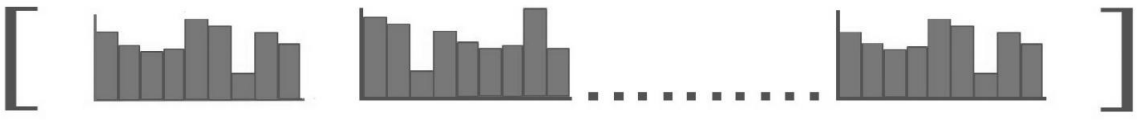


Fig. 5.6 Final feature vector

5.4 LINEAR SVM

Data can be clustered into 2 classes or groups by finding maximum marginal hyper plane that separates one class from another. In case of pedestrian detection, the classes correspond to positive samples(pedestrian) and negative samples(not a pedestrian) in a patch of the image .Margin is defined as the distance between the hyperplane and closest data point and the data points that lie on the boundary of the margin of the hyperplane are called support vectors .the computation of hyperplane is done using quadratic programming.

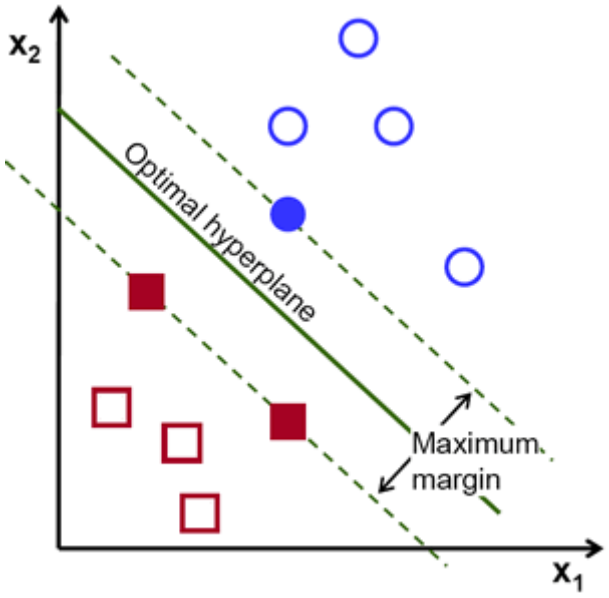


Fig. 5.7 Components of SVM [32]

5.5 NEED FOR NORMALISATION

Calculation of gradient:

	150	
100		60
	120	

Change in X direction:40 Change in Y direction:30

Gradient:40i+30j

Magnitude of gradient: 50 Orientation of gradient:$\tan^{-1}(30/40)$ =36.8°

Normalisation: gradient/magnitude= 0.8i+0.6j

Effect of increase in brightness by 50:

	200	
150		90
	170	

Change in X direction:40 Change in Y direction:30

Gradient:40i+30j

Magnitude of gradient: 50 Orientation of gradient:$\tan^{-1}(30/40)$ =36.8°

Normalisation: gradient/magnitude= 0.8i+0.6j (remains same)

Effect of increase in contrast by 1.5 :

	225	
150		90
	180	

Change in X direction:60 Change in Y direction:45

Gradient:60i+45j

Magnitude of gradient: 75 Orientation of gradient: $\tan^{-1}(45/60)$ =36.8°

Normalisation: 0.8i+0.6j (remains same)

So in the above example we see that by dividing the gradient vectors by their magnitude we can make them invariant to changes in contrast and brightness.

5.6 APPLICATION OF HOG

For Detecting of human beings accurately in a images. Here global feature is used to describe the entire person rather than collection of local features representing it in parts. Hog feature set performs equally well for other shape based object classes and can be used for car, table etc.

Fig. 5.8 Example of human detection through HOG

CHAPTER 6

EXPERIMENTS AND RESULTS

6.1 TRACKING EVALUATION MEASURES

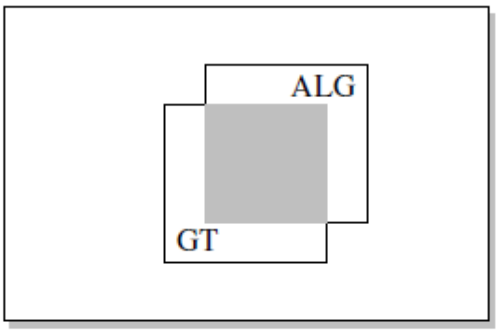
Evaluation involves comparison of bounding boxes obtained by tracking ,with ground truth data available with dataset. The result of comparison can be one of the following cases:

True positive-if amount of overlap between bounding boxes obtained by tracking and ground truth is greater than 50%.

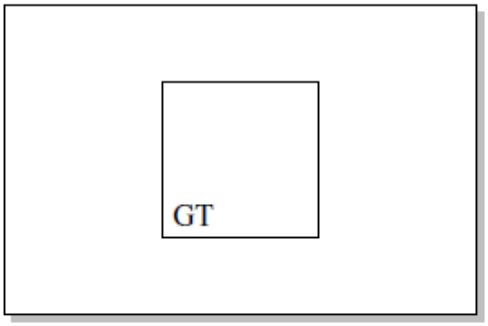
False negative- when identification and location of target is missed.

False positive- is the condition when object is indentified, that is bounding box is obtained by algorithm, but it is not actually a target as it is not present in the ground truth.

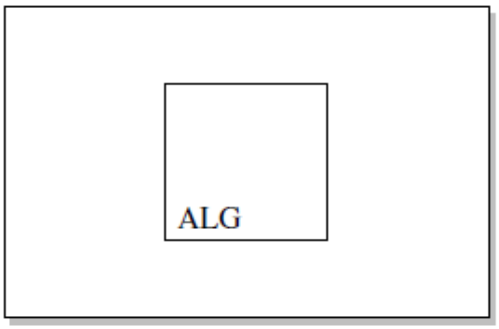
True negative- when the frame is neither an algorithmic output nor a part of ground truth, which means that target was not actually present at that location and it is correctly analysed.



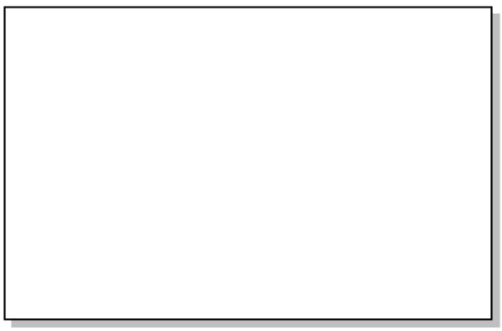
(a) True Positive



(b) False Negative



(c) False Positive



(e) True Negative

N_{tp} :number of true positives

N_{fp} : number of false positives

N_{fn} : number of false negatives

T^i : tracked bounding box in frame i

GT^i : ground truth bounding box in frame i

Based on the above values following performance matrices are calculated :-

1)Overlap

Ratio of intersection to the union of tracked bounding box and ground truth of the bounding box.[33]

$$(T^i \cap GT^i) / (T^i \cup GT^i) > 0.5$$

2)Precision

Fraction of positive examples that are correctly labelled .It signifies the usefulness of the search results.

$$N_{tp}/(N_{tp}+ N_{fp})$$

3)Recall

Fraction of examples that can be retrieved, also known as sensitivity. It signifies completeness of the results

$$N_{tp}/(N_{tp}+ N_{fn})$$

4) F-score

It is weighted average of precision and recall,with its best value of 1 and worst value of 0.

$$F=2. (Presision.Recall)/Presision+Recall$$

Precision and recall are standard matrices for performance evaluation. For good performance both of them are important.

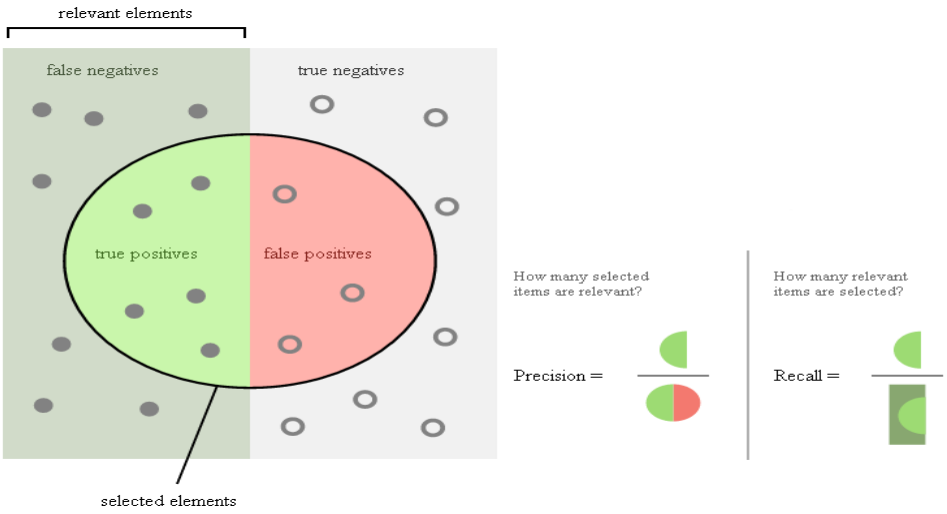


Fig 6.1 Calculation of precision and recall [34]

6.2 DATASET

A heterogeneous dataset is selected with image sequences possessing different tracking problems. Each dataset features the different issues faced during real time tracking such as , moving camera, occlusion, out of plane rotation etc. List of properties of each sequence is shown in table 1.

sequence	MVC	PO	FO.	IV	SC	BC	SO	OPR	DEF	FM
Pedestrian 3	yes	yes	yes	no	no	no	yes	no	no	no
Subway	no	yes	no	no	no	yes	no	no	yes	no
Jogging	no	yes	no	no	no	no	no	yes	yes	no
Human 7	yes	yes	no	yes	yes	no	no	no	yes	no

Table 1 Properties of image sequences

MVC: Moving Camera, PO: Partial Occlusion, FO: Full Occlusion, IV: Illumination Variation, SC: Scale Change, BC: Background Clutter, SO: Similar Object, OPR: Out of Plane Rotation, DEF: Deformation, FM: Fast Motion [35][1]

6.3 DISCUSSION OF RESULTS

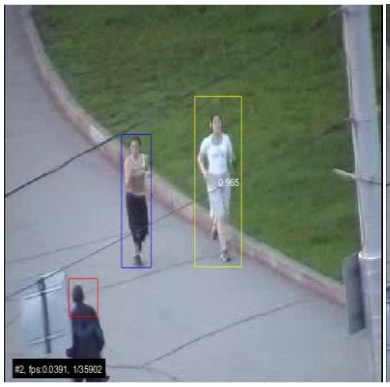
The proposed algorithm tracks multiple pedestrians present in sequence Jogging and Pedestrian 3 as shown in Figure 6.2 and Figure 6.3. Figure 6.2 tracks 2 objects marked by yellow and blue bounding boxes. Both the objects becomes partially occluded after frame 45 due to presence of street light in their path .But after street light is crossed their bounding boxes is regained. This shows that the algorithm is robust to partial occlusion.

Figure 6.3 shows extension of TLD to track 3 objects in sequence 'Pedestrian 2' of TLD dataset. The pedestrians are marked with red, blue and yellow bounding box. Here pedestrian in yellow bounding box goes through occlusion after frame 30 but it is redetected as soon as it enters the scene again at frame 41.This shows that algorithm is robust to full occlusion as well. Occlusion handling can also be observed in sequence 'Pedestrian 3' shown in Fig. 6.4 .The pedestrian, represented by yellow bounding box is under partial occlusion at frame 53 and then full occlusion after frame 55, but it is redetected at 81 as soon as it re-enters the frame.

The frame rate is varies slightly in different runs as the algorithm randomly forgets the stored templates in the database randomly when the number of templates is higher than threshold. So it is calculated by taking average over 3 runs. As we can see from table 2, the frame rate is less than or equal to the real time camera frame rate of 30 frames per second.

Figure 6.6 shows the trajectory of a pedestrian in sequence 'Pedestrian3'. Figure 6.7 shows The P and N experts used in the learning step. P–experts shown on right side and N-expert ,are shown in left side of the figure .

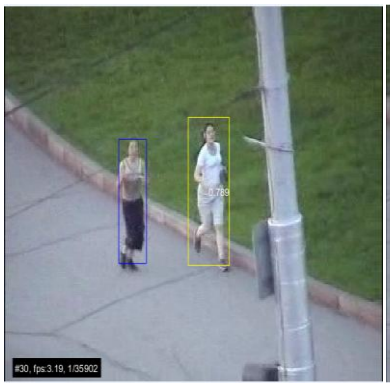
The first column of Table 3 shows the total number of frames in the sequence and second column shows the number of frames tracked by using the discussed in section 6.1. The initial bounding box used for the first frame, was provided along within the dataset. Some of the frames could not be tracked due to full occlusion, as then the object is not visible to the camera in that scene. Last 3 columns gives the evaluation measures of precision, recall and F-score for single object tracking case. These value are >0.85 in all the cases and obtain maximum F-score of 0.98 .This shows that the algorithm is robust to various difficulties of tracking present in each sequence, given in table 1.



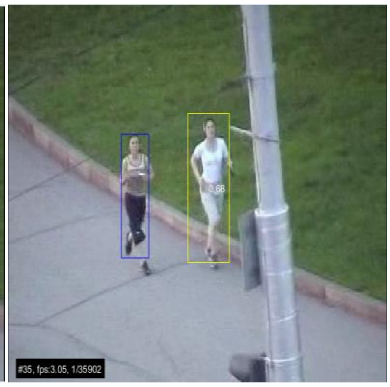
Frame 2

Frame 10

Frame 20



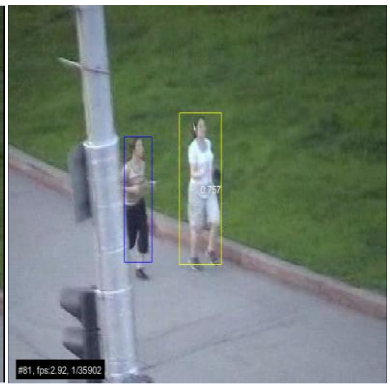
Frame 30



Frame 35

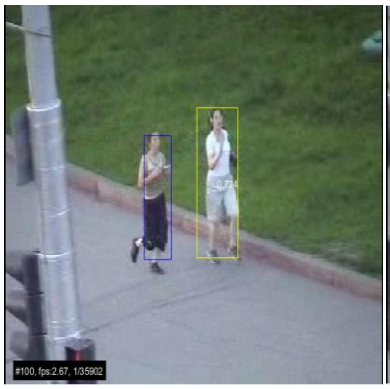
Frame 45

Frame 64

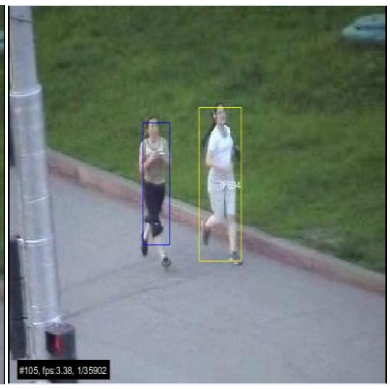


Frame 81

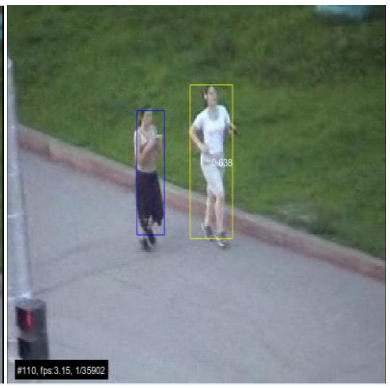
Frame 88



Frame 100



Frame 105



Frame 110

Fig 6.2 Tracking of 2 pedestrians in sequence 'jogging'.

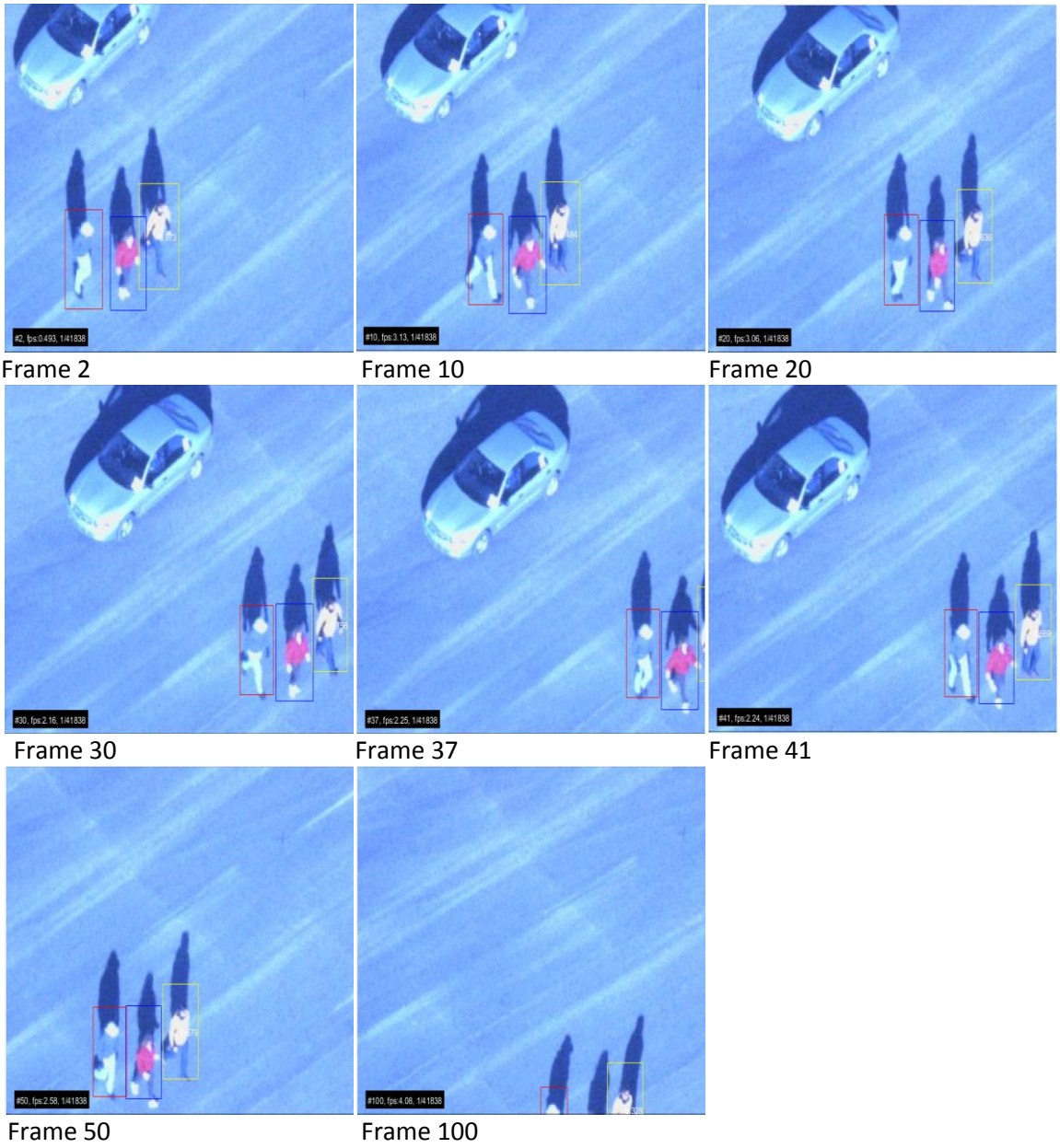


Fig 6.3 Tracking of 3 pedestrians in sequence 'Pedestrian 2'.

sequence	Frame rate (frames per second)
Pedestrian 3	10.00
Subway	8.95
Jogging	8.89
Human 7	12.07

Table 2 Execution time

Fig 6.4 Occlusion handling in sequence ‘Pedestrian 3’.

sequence	Number of frames	Number of frames tracked successfully	Precision	Recall	F-score
Pedestrian 3	184	156	0.975	1.00	0.98
Subway	175	157	0.897	0.89	0.89
Jogging	307	283	0.92	0.92	0.92
Human 7	250	224	0.976	0.97	0.97

Table 3 Evaluation of sequences

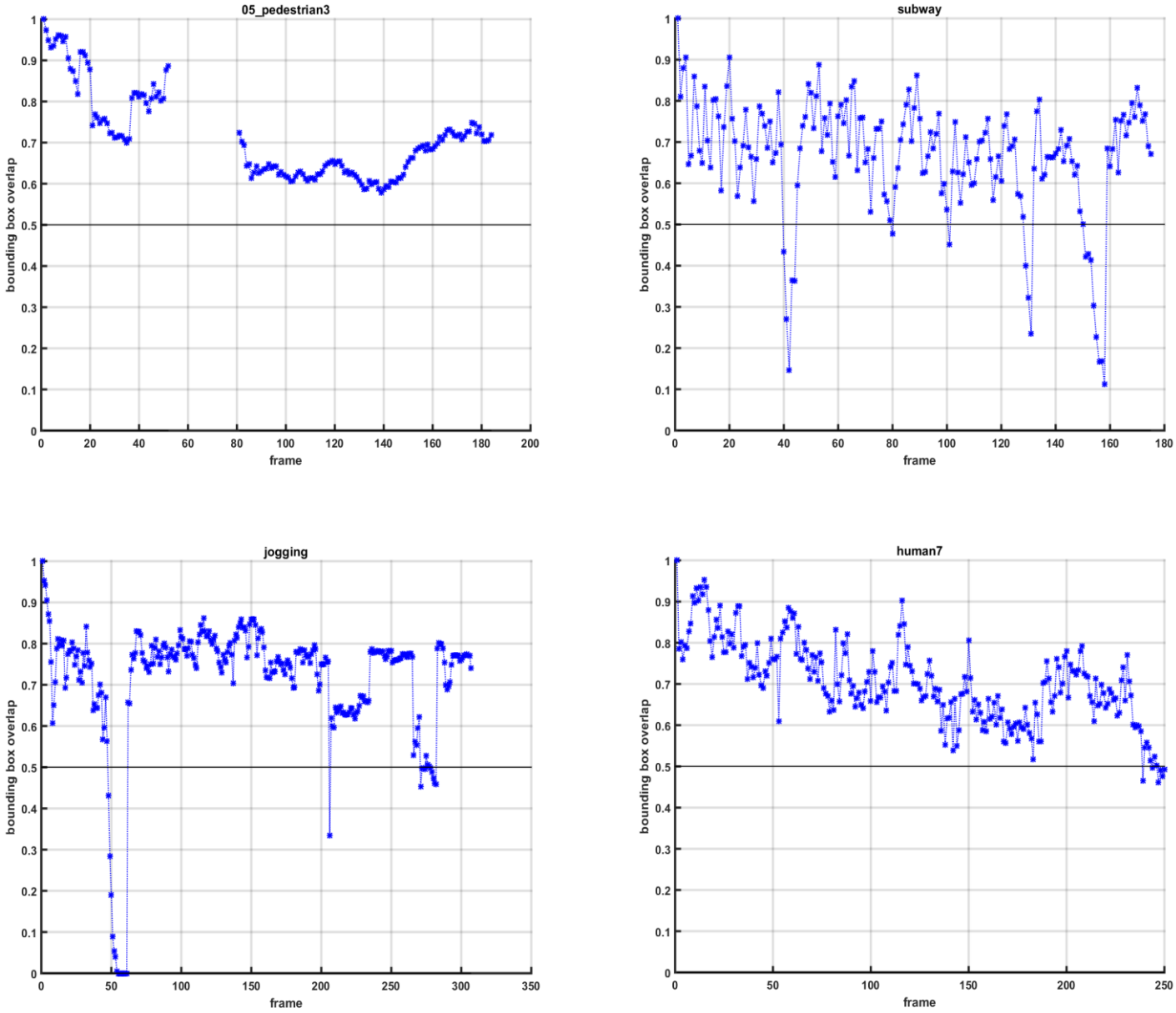


Fig 6.5 Overlap of bounding box and ground truth

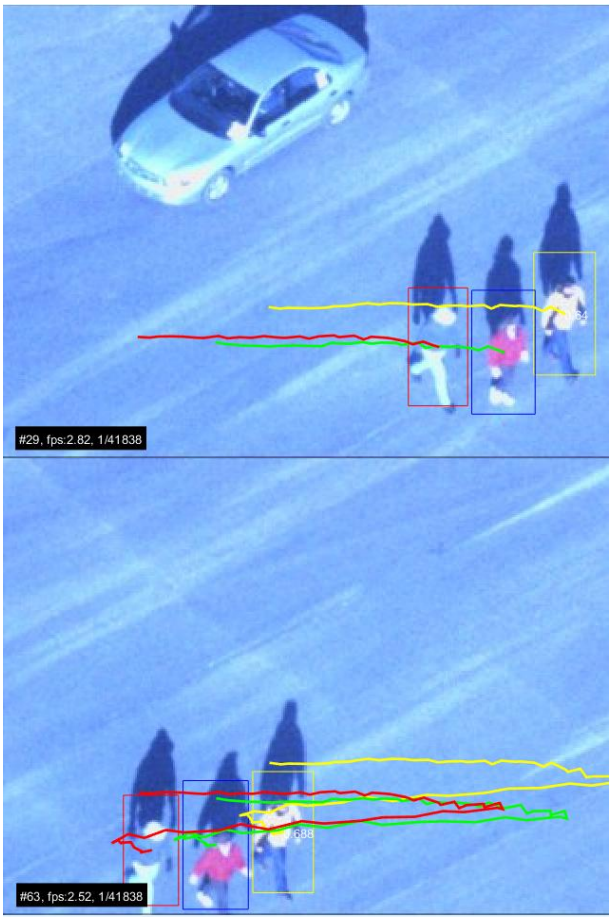


Fig6.6 Trajectory of pedestrians in sequence 'Pedestrian3'

Fig 6.7 N experts representing background and P-experts representing target object

CHAPTER 7
CONCLUSION AND FUTURE SCOPE

The proposed algorithm is able to track multiple pedestrians and their trajectories successfully. It can be further used for Surveillance applications such as crowd flux analysis and crowd management.

Although HOG significantly outperforms many available pedestrian detection methods, other person detector can be used such as convolution neural networks [36] for improved results.

Instead of passing RGB images we can pass 3D depth data from Kinect sensor as input to the algorithm for improved results.

Tracking and learning component of the algorithm can be run in parallel for faster speed. Speed can also be improved by using GPU acceleration.

TLD can be extended to multi camera scenario. Information about the orientation of objects could be retrieved by employing an affine transformation model for the Lucas-Kanade tracker [11].

REFRENCES

[1] Z.Kalal, K.Mikolajczyk, and J.Matas,"Tracking-learning-detection IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 34, no. 7,pp. 14091422, Jul. 2012.

[2] Harris, Chris, and Mike Stephens. "A combined corner and edge detector."Alvey vision conference. Vol. 15. 1988.

[3] Dalal, N. and B. Triggs. "Histograms of Oriented Gradients for Human Detection", IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Vol. 1 (June 2005), pp. 886893.

[4]Yilmaz, Alper, Omar Javed, and Mubarak Shah. "Object tracking: A survey."*Acm computing surveys (CSUR)* 38.4 (2006): 13.

[5] Ballard, Dana H., and Christopher M. Brown. "Computer Vision, article, 4 pages Prentice-Hall." *Englewood Cliffs, New Jersey, believed to be published more than one year prior to the filing date of the present application* (1982).

[6] Zhu, Song Chun, and Alan Yuille. "Region competition: Unifying snakes, region growing, and Bayes/MDL for multiband image segmentation." *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 18.9 (1996): 884-900.

[7] Paragios, Nikos, and Rachid Deriche. "Geodesic active regions and level set methods for supervised texture segmentation." *International Journal of Computer Vision* 46.3 (2002): 223-247.

[8] Elgammal, Ahmed, et al. "Background and foreground modeling using nonparametric kernel density estimation for visual surveillance." *Proceedings of the IEEE* 90.7 (2002): 1151-1163.

[9] Comaniciu, Dorin, Visvanathan Ramesh, and Peter Meer. "Kernel-based object tracking." *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 25.5 (2003): 564-577.

[10] Tuytelaars, Tinne, and Krystian Mikolajczyk. "Local invariant feature detectors: a survey." *Foundations and Trends® in Computer Graphics and Vision* 3.3 (2008): 177-280.

[11] Bouguet, Jean-Yves. "Pyramidal implementation of the affine lucas kanade feature tracker description of the algorithm." Intel Corporation 5.1-10 (2001): 4.

[12] Horn, Berthold KP, and B. G. Schunck. "Determining optical flow-a retrospective." *Artif. Intell. v59* (1994): 81-87.

[13] Mikolajczyk, Krystian, and Cordelia Schmid. "Scale & affine invariant interest point detectors." *International journal of computer vision* 60.1 (2004): 63-86.

[14] Bay, Herbert, Tinne Tuytelaars, and Luc Van Gool. "Surf: Speeded up robust features." *Computer vision–ECCV 2006*. Springer Berlin Heidelberg, 2006. 404-417.

[15] Oh, Songhwai, Stuart Russell, and Shankar Sastry. ”Markov chain Monte Carlo data association for general multiple-target tracking problems.” 43rd IEEE Conference on Decision and Control, 2004. CDC. Vol. 1. IEEE,2004.

[16] Y. Li, C. Huang and R. Nevatia, ”Learning to associate: HybridBoosted multi-target tracker for crowded scene,” IEEE Conference on Computer Vision and Pattern Recognition, 2009. Miami, FL, 2009, pp. 2953-2960.

[17] M. D. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier and L. V. Gool, ”Robust tracking-by-detection using a detector confidence particle filter,” IEEE 12th International Conference on Computer Vision, Kyoto, 2009, pp.1515-1522.

[18]J. Berclaz, F. Fleuret and P. Fua, ”Multiple object tracking using flow linear programming,” Twelfth IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETS-Winter), 2009, Snowbird, UT, 2009, pp. 1-8.

[19] Smeulders, Arnold WM, et al. "Visual tracking: An experimental survey."*Pattern Analysis and Machine Intelligence, IEEE Transactions on* 36.7 (2014): 1442-1468

[20] Z. Kalal, J. Matas and K. Mikolajczyk, ”P-N learning: Bootstrapping binary classifiers by structural constraints,” IEEE Conference on Computer Vision and Pattern Recognition (CVPR), San Francisco, CA, 2010, pp. 49- 56. doi: 10.1109/CVPR.2010.5540231

[21] Z. Kalal, K. Mikolajczyk and J. Matas, ”Forward-Backward Error: Automatic Detection of Tracking Failures,” 20th International Conference on Pattern Recognition (ICPR), Istanbul, 2010, pp. 2756-2759. doi: 10.1109/ICPR.2010.675

[22] B.D. Lucas, T. Kanade, "An Image Registration Technique with an Application to Stereo Vision", in Proceedings of Image Understanding Workshop, 1981, pp. 121-130.

[23] H. Schneiderman, ”Feature-centric evaluation for efficient cascaded object detection,” Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR , 2004, pp. II-29-II-36 Vol.2. doi: 10.1109/CVPR.2004.1315141

[24] P. Viola and M. Jones, ”Rapid object detection using a boosted cascade of simple features,” Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition,CVPR , 2001. pp. I-511-I-518 vol.1.

[25]P. Viola and M. Jones, “Rapid Object Detection Using a Boosted Cascade of Simple Features,” Proc. IEEE CS Conf. Computer Vision and Pattern Recognition, 2001.

[26] V. Lepetit, P. Lagger and P. Fua, ”Randomized trees for real-time keypoint recognition,” IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05), 2005, pp. 775-781 vol. 2. doi: 10.1109/CVPR.2005.288

[27] slides from Derek Hoiem ,”computer vision”CS 543/ECE549 ,University of Illinois.

[28] “slides matching with invariant features” –Darva florova ,Denis Simakov ,the Weizmann institute of Science,march 2004

[29]Robert Collins slides- CSE486 ,penn state

[30] Matching with Invariant Features Darya Frolova, Denis Simakov The Weizmann Institute of Science March 2004

[31]Local Image Features by Jason Corso ,College of Engineering ,Electrical engineering and Computer science ,University of Michigan.

[32]Opencv documentation
http://docs.opencv.org/2.4/doc/tutorials/ml/introduction_to_svm/introduction_to_svm.html

[33]Everingham, Mark, et al. "The pascal visual object classes (voc) challenge."*International journal of computer vision* 88.2 (2010): 303-338.

[34] Wikipedia page : https://en.wikipedia.org/wiki/Precision_and_recall

[35]Dataset taken from Wu, Yi, Jongwoo Lim, and Ming-Hsuan Yang. "Online object tracking: A benchmark." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2013.
http://cvlab.hanyang.ac.kr/tracker_benchmark/benchmark_v10.html

[36] M. Szarvas, A. Yoshizawa, M. Yamamoto and J. Ogata, ”Pedestrian detection with convolutional neural networks,” IEEE Proceedings. Intelligent Vehicles Symposium, 2005. pp. 224-229. doi:10.1109/IVS.2005.1505106