Indian Statistical Institute
Kolkata

M.Tech. (Computer Science) Dissertation

Detection and Tracking of Humans in Video

A dissertation submitted in partial fulfillment of the requirements
for the award of Master of Technology in
Computer Science

Author:
Sanjana Sinha
Roll No: MTC 1321

Supervisor:
Dr. CA Murthy
Machine Intelligence Unit

M.TECH(CS) DISSERTATION
THESIS COMPLETION CERTIFICATE

Student : Sanjana Sinha (MTC1321)
Topic : Detection and Tracking of Humans in Video
Supervisor : Dr. CA Murthy

This is to certify that the thesis titled "Detection and Tracking of Humans in Video" submitted by Sanjana Sinha in partial fulfillment for the award of the degree of Master of Technology is a bonafide record of work carried out by her under my supervision. The thesis has fulfilled all the requirements as per the regulations of this Institute and, in my opinion, has reached the standard needed for submission.

CA Murthy

Date : 10^{th} *July*, 2015

Abstract

This objective of this thesis is to detect the presence of moving people and count the number of individuals in a given video. The problem is to identify motion regions in the video, separate the motion regions from the static background, form foreground objects from the motion regions, classify each object to determine if it is a person, then track each person detected throughout the duration of the video.

Our focus is to improve the classification accuracy by using new features that are based on pairwise distance between training points in feature space, and reduce the dimensionality of the feature space by applying a data condensation method prior to feature extraction. The proposed method gives better accuracy and lower misclassification rate on a benchmark image dataset, as well as a higher tracking accuracy on video datasets.

Acknowledgement

I would like to thank my dissertation advisor, Dr. CA Murthy for his guidance and support throughout the duration of my dissertation.

List of Figures

Contents

Chapter 1

Introduction

1.1 Background and Motivation

The problem of detecting human presence and counting the number of moving people in video has been a widely studied research topic and yet its many challenges still leave room for improvements over existing solutions. The problem has three aspects - moving object detection, recognition of persons among the detected objects, and keeping track of the number of distinct people detected. Each of these three stages have its own challenges that have been addressed to varying extent by state-of-the-art methods. The focus of this thesis is on understanding the challenges and improving the accuracy of the problem solution by using a different approach to human detection in video.

Human detection in video and analysis of human motion has a wide number of applications, such as visual surveillance for abnormal event detection, crowd counting in a mall entrance, individual person identification and automatic pedestrian detection systems. Each application involves a separate context and a different set of challenges for the person detection and tracking systems. For example, visual surveillance typically involves stationary camera video, taken from multiple cameras each facing a different angle. Advanced Pedestrian detection systems employ an on-board camera on a vehicle, thereby restricting the use of motion detection techniques that are applicable in case of video obtained via static camera. Counting people at a mall entrance requires stationary cameras, often positioned at different angles. Each application governs it's own set of rules and constraints for detection and tracking. In this thesis the focus is on video obtained through a stationary camera, such as surveillance video.

The three stages of the online method for human detection and tracking are:

1. **Background subtraction/Foreground segmentation :** The foreground

comprising of the motion regions is separated from the stationary background pixels in each video frame. The moving objects are obtained by noise removal, elementary image morphological operations followed by a connected components algorithm on the foreground pixels.

2. **Classification :** The moving objects are represented using appropriate features, which are needed for binary classification into the human and non-human class. Training images from both the class of humans and the complementary class are used to train a detector.

3. **Tracking :** Tracking involves find a temporal correspondence between the detected object in the each video frame. Each individual is associated with a predicted trajectory which is dynamically updated on the basis of actual observations. At the end of the tracking phase the total number of distinct individuals in a video is obtained.

There are many challenges surrounding the above mentioned three stages of the person detection and tracking system, some of which are elaborated below:

- The human form as visible in video exhibits high variation due to difference in dimensions, attire, pose, angle of view (front, rear or side view), and sometimes carrying different objects, make the process of identification as human quite difficult. Moreover the basic assumption of fully visible people is mandatory for the human detection process.

- The presence of background clutter, slow-moving background objects, change in illumination, presence of shadow, obstruction from view (occlusion) makes the process of separating background from foreground objects difficult. Since there can be small regions of background intensity variations which might be falsely labelled as foreground, a threshold on the area of the foreground object is used in order to prevent false detections due to pixel distortions and video compression.

- A person detected in a frame may not be visible in subsequent frames (due to occlusion), and again reappear in view after a time lapse. The people may move together as a group, split, and merge again. This makes the process of counting difficult. So a threshold is necessary on the time duration of a trajectory (path) of a detected object for updating of person count.

1.2 Literature survey

1.2.1 *Background Subtraction*

Foreground segmentation is the method for detection of the moving objects of interest (humans) from the rest of the rest of the video frame (background). Background subtraction is a useful method for foreground segmentation, when the video is obtained via a stationary camera. The background image or "background model" is a representation of the scene with no moving objects and must be dynamically updated to adapt to the change in illumination.

The simplest approach for background subtraction is to model the intensity value at pixel location in a video frame as a Gaussian distribution, whose parameters are updated using dynamically [27]. This method is suitable for indoor environments where the background is unimodal, and illumination is constant.

To handle effects of varying illumination and multimodal backgrounds, typical of outdoor environments, pixel intensity as a function of time is modelled as a mixture of Gaussians, whose parameters are updated online to meet varying conditions of illumination, and objects moving in and out of the video frames. This method by Stauffer and Grimson [24] has become the standard of background subtraction.

The above methods of background subtraction are parametric in nature. A non-parametric approach by Elgammal et al.[8] uses a kernel density estimation of the pixel intensity density function. This technique performs better than MoG when there are rapid changes in the background, the Kernel based method adapts much more quickly to variations in the background. But the computational requirements are much higher for this method.

1.2.2 *Human detection*

Detecting moving people in video is one of the most challenging tasks in object detection, mainly due to the variety in shape, size, colour and posture of each person. The object classification system receives a list of Regions of Interest (ROIs) from the result of the background subtraction system, and each such ROI is subject to binary classification to determine whether the object it contains is human or not.

One of the earliest person detectors was the Chamfer System, a silhouette-matching algorithm proposed by Gavrila et al. [11] [10] [12]. This system consists of a hierarchical template-based classifier that matches distance-transformed ROIs with template shapes or exemplars. Another shape-based approach to detection is by Wu and Nevatia [28], which utilizes a large pool of short line and curve segments,

called “edgelet” features, to represent shape locally. Similarly, “shapelets” [22] are shape descriptors in local patches. Multiple shapelets are combined into an overall detector by Boosting.

Papageorgiou et al. [20] carried out person detection using Haar wavelets as features to train a quadratic support vector machines (SVMs). The feature space consisting of overcomplete dictionary of multiscale Haar wavelets captures the significant information about elements of the class of humans. The SVM classifier is trained using positive and negative examples from the class of humans and class of all non-human objects respectively. While the Haar wavelet based detector was proposed for static images, an efficient moving person detector was built by Viola et al.[25]. This method uses Haar-like features that incorporates spatial as well as motion information, and Adaboost cascades for classification.

A significant improvement to person detection results was obtained by adopting the use of gradient-based features, over intensity-based features. Dalal and Triggs [6] [7] present a human classification scheme that uses SIFT-inspired [15] features, called histograms of oriented gradients (HOGs), and a linear SVM classifier. Zhu et al. [33] use HOG features with an additional constraint of integral histograms for fast computation, and feature selection using Adaboost cascades.

While no single feature has been shown to outperform HOG, additional features can provide complementary information. Wang et al. [26] combined a texture descriptor based on local binary patterns (LBP) [18] with HOG with a slightly modified linear SVM classifier.

1.2.3 *Tracking*

Tracking is the temporal correspondence between the individual persons detected in the current frame with those in the previous frame. Each such correspondence is a temporal trajectory in state space. The detected objects can be represented as a set of points, in which case the tracking mechanism is referred to as point tracking. Statistical correspondence methods can solve point tracking problems by taking the measurement and the model uncertainties into account during object state estimation. Bayesian filters such as Kalman filter [21], Extended Kalman filter [3] and Particle filters [23] are widely used for single object tracking, which can be extended to multiple object tracking with the help of data association to assign detections to target tracks.

A Kalman filter is used to estimate the state of a linear system where the state is assumed to be distributed by a Gaussian. Extended Kalman filter relaxes the assumption of linearity in state equations, thus resulting in a sub-optimal solution.

A particle filter, on the other hand is well-suited for state estimation when the state transition equations are non-linear and the state variables do not follow Gaussian distribution. There has been extensive application of particle filters in single object tracking [14] as well as multi-target tracking [30] [13] [29] [31] [19] [4] [5] [16].

1.3 Approach Overview

The existing literature separately tackle the three problems - background subtraction, human detection (by classification) and human tracking. Our approach integrates all three parts into one, solely for the purpose of counting the total number of moving people in a video scene. The background subtraction method used in our implementation is one of the state of the art algorithms, with slight adjustment of parameters and post-processing to adapt to the visual conditions of our test video datasets.

The human detection stage involves a feature extraction phase followed by classification. The most popular feature space for human detection is Histogram of Oriented Gradients(HOG), which is used in some form in almost every modern implementation of person detectors. This feature space when used with a classifier such as linear SVM, results in a number of misclassifications even using the benchmark training datasets. Our approach involves a pairwise distance-based transformation of the HOG features which when used to train the classifier (linear SVM) gives improved accuracy of classification in terms of lower misclassification rate, thereby improving the results of human detection. Since the input to the tracking stage is the detected persons resulting from the human detection stage, this also results in more robust tracking and accuracy of people counting.

Our tracking stage uses the Bayesian particle filter, updated with our own observation model, with additional rules to adapt to a multiple-target tracking framework. The direct result of the tracking stage is the count of the individual persons in a video scene.

1.4 Organization of the Report

This Chapter introduced the problem of human detection and tracking in video, listed the main applications, mentioned some of the challenges, outlined the main steps of the algorithm, described previous related research work and provided a basic overview of our approach and the motivation behind it. The remaining chapters are organized as follows:

- **Chapter 2** gives a description of the background subtraction algorithm used.
- **Chapter 3** provides a description of the proposed human detection algorithm, using distance-transformed features and an SVM classifier to detect the moving people in the the video.
- **Chapter 4** gives the experimental results of the proposed method and its comparison with the state of the art on benchmark image datasets.
- **Chapter 5** describes the algorithm for tracking and defines rules and conditions for data association in a multi-person tracking framework.
- **Chapter 6** gives the experimental results on benchmark video datasets using both the proposed method and existing method for feature extraction.
- **Chapter 7** summarises our approach and results along with the future scope.

Chapter 2

Background Subtraction

In order to detect the moving objects in a video, each video frame must be segmented into the foreground, consisting of moving object pixels, and the background, consisting of stationary portions of the video frame. An estimate of the background, often called a background model is computed and evolved frame by frame, moving objects in the scene are detected by the difference between the current frame and the current background model. The background is represented as a statistical model, in the form of either a single Gaussian distribution, or a mixture of Gaussians. Using a single gaussian to model the background suffers from a lot of drawbacks, such as failure to represent multiple objects, inability to adapt to changing illumination. On the other hand, a mixture of Gaussians, whose paramaters are dynamically updated, can handle some of the issues, such as multiple objects in foreground, background clutter, multimodal background, slow changes in illumination. This method, by Stauffer et al. [24] is the most popular background subtraction algorithm used by different applications.

2.1 Background Modelling using Mixture of Gaussians

Intensity value $I_{x,y}$ at a pixel position (x, y) over a set of frames at time $t = t_1, t_2, \cdots, t_{n-1}$ is assumed to have a mixture density function of K Gaussians, where K is the number of number of surfaces visible at pixel position (x, y) over the duration of the the video. K is assumed to be a constant.

A pixel process is a time series of intensity values at a particular pixel location.

At a time instant t, pixel process at pixel (x_0, y_0) is

$$\{X_1, X_2, \cdots, X_t\} = \{I(x_0, y_0, i) : 1 \leq i \leq t\}$$

The pixel process at time t is modeled by a mixture of K Gaussian distributions

$$P(X_t) = \sum_{i=1}^{K} \omega_{i,t} * \eta(X_t, \mu_{i,t}, \Sigma_{i,t}) \tag{2.1}$$

$P(X_t)$ is the mixture density function, where the each of the individual multivariate normal distributions with mean μ_k and covariance matrix Σ_k have the density function of the following form:

$$\eta(X_t|\mu, \Sigma) = \frac{1}{(2\Pi)^{\frac{n}{2}}|\Sigma|^{\frac{1}{2}}} \exp^{-\frac{1}{2}(X-\mu)^T\Sigma^{-1}(X-\mu)} \tag{2.2}$$

For computational reasons, the covariance matrix is assumed to be of the form

$$\Sigma_{k,t} = \sigma_k^2 I \tag{2.3}$$

The weights of the gaussians are normalized ($\sum_{i=1}^{K} \omega_i = 1$)

2.2 Dynamic updation of parameters

If the pixel process were a stationary process, a standard method for maximizing the likelihood of the observed data is expectation maximization (EM) algorithm. However, since new objects are introduced in the video frames and previously observed objects leave the frames, an on-line approximation to EM algorithm was proposed by [24].

If $X_t = I_{x,y} = (i_r, i_g, i_b)$ is the intensity vector at a pixel location (x,y), X_t is assigned to one of the K Gaussian distributions by matching the parameters of the distribution to the observed value X_t, and thereafter the parameters of the mixture distribution is updated accordingly.

A match is said to occur between X_t and the kth Gaussian distribution if

$$|I_{x,y,j} - \mu_{k,j}| < 3\sigma_k, j \in \{r, g, b\}, \exists k \in K \tag{2.4}$$

Note that if there is more than one match, in our implementation we have assigned it to the Gaussian of the highest weight.

The weights of the Gaussians are updated as follows:

$$\omega_k = \omega_k + \alpha(1 - \omega_k) \tag{2.5}$$

$$\omega_j = (1 - \alpha)\omega_j, j \in K, j \neq k \tag{2.6}$$

where k is the matched distribution and α is a learning rate

The parameters of the matched distribution (kth Gaussian) are updated as follows:

$$\mu_{k,t} = (1 - \rho)\mu_{k,t} + \rho X_t \tag{2.7}$$

$$\sigma^2_{k,t} = (1 - \rho)\sigma^2_{k,t} + \rho(X_t - \mu_{k,t})^T(X_t - \mu_{k,t}) \tag{2.8}$$

where $\rho = \alpha\eta(X_t|\mu_k, \Sigma_k)$

When there is no match with any Gaussian, i.e, the condition defined in 2.4 fails $\forall k \in K$, the weights of the Gaussians are updated.

$$\omega_j = (1 - \alpha)\omega_j, \forall j \in K, \tag{2.9}$$

After that, the distribution with the least weight is replaced with a new distribution having the following parameters.

$$\mu_k = X_t$$

$$\sigma^2_k = \sigma^2_0$$

$$\omega_k = \omega_0$$

where σ^2_0 is an initial value of high variance, set experimentally, and ω_0 is an initial low value of weight assigned to the new distribution.

A matched distribution implies that the pixel intensity value X_t is previously observed and assigned to one of the K Gaussian distributions. An unmatched value of X_t implies it either belongs to a new object introduced to the frame, or was previously replaced due to low weight of the distribution it was initially assigned to. A high value of K implies more objects can be represented in the mixture density function, but with additional computational cost. The original paper uses a value of $K = 3$ to 7.

2.3 Background Model Estimation

From the mixture of Gaussians obtained in the previous stage, each Gaussian must be evaluated to determine whether the pixel process it models belong to a foreground

object or the background. For this reason, there is a basic assumption in this method that the background is visible for the greater portion of time than the foreground objects.

The Gaussians are ordered by the value of ω/σ. Since the background intensity levels will have lower variation than object pixels, and the assumption that the background being visible for a longer time results in higher weight of the Gaussians representing background pixel processes. The list of Gaussians ordered by decreasing value of ω/σ will have the more predominant background distributions in the beginning and the less frequent transient background distributions near the end of the list, followed by the moving object distributions.

From the ordered list the first B distributions are chosen to represent the Background model, where

$$B = \underset{b}{\operatorname{argmin}}(\sum_{i=1}^{b} \omega_k > T) \tag{2.10}$$

where T is a threshold that indicates the minimum portion of the frame that is occupied by the background.

Let k is the index of the matched/replaced Gaussian for pixel intensity X_t. If k belongs to the first B ordered distributions, the the pixel at location (x, y) is marked as background at time instance t. Otherwise the pixel is marked as object pixel at time instance t.

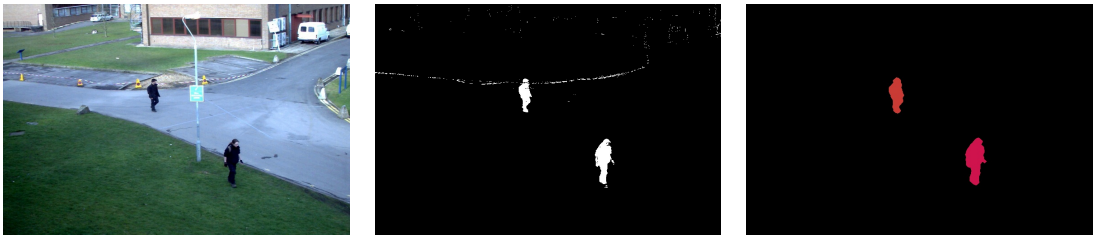
2.4 Parameter Initialization

In our implementation, the parameters of the algorithm have been initialized with experimentally determined values.

- Number of Gaussians $K = 7$. It was observed that a lower number of Gaussians does not allow much variation in the background model,whereas a higher value of K results in unnecessary computational overhead without any significant improvement to performance.

- Learning Rate $\alpha = 0.01$. A higher learning rate results in portions of slow-moving objects becoming part of the background quickly. A lower learning rate results in very low adaptability to changes in pixel values over time.

- Threshold $T = 0.7$. If the threshold is lower, a number of background pixels showing slight intensity variations is detected as object pixels. A higher value of T results in slow-moving object pixels becoming part of background.



(a) (b) (c)

Figure 2.1: (a) is the original video frame, (b) is the result of background subtraction, (b) is after noise removal and connected components analysis

2.5 Noise Removal and Connected Components Analysis

In each video frame the background subtraction results in some pixels being marked as foreground and the rest as background. In order to obtain the distinct objects made up of foreground pixels, it is necessary to carry out connected components analysis on the foreground pixels. The resultant video frame consists of distinct connected components, each of which represents a moving object. Each such connected component represents a Region of Interest (ROI), which is used by the classification stage.

Background subtraction in general results in a number of background pixels being falsely labelled as object, due to repetitive background motion or poor video quality as a result of video compression. In such cases, often a single pixel or a group of two or three pixels is identified as a single object after connected component analysis. On other instances, a portions of a foreground object is labelled as background due to shadows, insufficient illumination, or colour match of garments with the background. In such cases a single foreground object is split into a number of separate connected regions.

In order to reduce such anomalies, median filtering for noise removal and binary dilation for connecting nearby disconnected components has been performed prior to the connected component analysis. Figure 2.1 shows the results of the background subtraction and the post-processing.

Chapter 3

Human Detection by Classification

After obtaining the segmented foreground objects in the form of bounding rectangles around each moving objects, also termed as ROIs, the task is to classify each such ROI either into the class of all humans or into the class comprising of all objects other than human beings. This binary classification is a challenging task mainly because the variety of human body shape, orientation, colour of garments. The features used for classification must represent the generic representation of the human body which is mostly invariant of scale change, colour and illumination. Pixel intensity values, which is a commonly used feature in face recognition problems, cannot be used as a feature for human detection because of the problems mentioned above. Pixel gradient based descriptors capture the contour of the human body better than than intensity-based descriptors, but the location of gradients is subject to rotation and translation, due to the hand and leg movements. To overcome these challenges, Histogram of Oriented Gradients (HOG) features were proposed by Dalal and Triggs [6] and no single feature has been shown to outperform HOG in human detection till date.

3.1 State of the art: Histogram of Oriented Gradients Features

The idea behind Histogram of Oriented Gradients (HOG) originated from Edge-Orientation Histograms [9] and SIFT [15]. The HOG feature space encodes several information that intensity or frequency-based methods cannot encode. HOG features capture edge and contour information in the form of image gradients. Since the human body contour is subject to changes, the feature space must be invariant to rotation and translation to effectively represent the human shape. For this

purpose, instead of recording raw image gradients, HOG encodes information in the form of orientation and spatial histograms over a grid of cells. Each orientation histogram bin corresponds to a range of gradient directions in order to permit movement of human limbs. The invariance to translation of human body segments is captured by the spatial histogram bins. The HOG feature computation is provided below.

- The ROI image is resized to a fixed size that matches the size of the detection window, whose default size is 128x64.
- The detection window is divided into multiple overlapping blocks of 16x16 pixels. The horizontal and vertical overlap between adjacent blocks is 8 pixels wide.
- Each such block is divided into 4 cells of size 8x8 each. Thus each cell has 4-fold coverage by blocks.
- Each cell contains an orientation histogram of the image gradient directions within the boundaries of the cell. Each histogram bin represents a range for gradient directions, and voting into a bin is based on gradient magnitude.
- Fine orientation binning is performed within each cell, and course spatial binning is performed within a block of several cells.
- The histogram bins are normalized over each cell, and then over the entire block.
- The histogram entries from all the overlapping blocks are combined to form the HOG descriptor feature vector.

For a detection window of size 128x64, with block size 16x16, and 8 pixel block overlap, there are a total of 105 blocks. Each block containing 4 cells and 9 histogram bins per cell makes a total feature size of 3780, for a single test image ROI.

3.2 Proposed Distance-based Feature Transformation

The performance of the binary classification depends on how well separated are the points belonging to one class from the points in the other class. A transformation

is proposed for the input HOG feature space, the resulting feature space providing greater separation between the two classes. The transformed feature space encodes the pairwise distance between every two training points in the HOG feature space as a new feature. The total number of dimensions is equivalent to the total number of training points. When a test point is to be classified, its distance from each of the training points forms the new feature vector. A similar method of constructing features using pairwise distances to cluster centres was proposed in [32] to solve the problem of multi-label learning. The intuition behind this proposed feature extraction method is that the distance between any point to the rest of the points in the same class is lesser than its distance to points from the other class.

3.2.1 *Proposed Algorithm*

The proposed algorithm works as follows:

Let $S = \{\underline{x}_1, \underline{x}_2, \cdots \underline{x}_N\} \subseteq \mathbf{R}^d$ be the set of points in the input feature space.

Let θ_i denote the class label of $\underline{x}_i$, $i = 1, 2, \cdots, N$.

$\theta_i \in \{1, -1\}\ \forall i = 1, 2, \cdots, N$

Let there be m training examples from the positive class, i.e, the class of all images of humans. Let n be the number of examples from the negative class, comprising of images of objects other than humans.

The positive training points in the input feature space are

$$S_1 = \{\underline{x}_i \in S : \theta_i = 1, i = 1, 2, \cdots, N\}$$

Similarly the negative training points are

$$S_2 = \{\underline{x}_i \in S : \theta_i = -1, i = 1, 2, \cdots, N\}$$

Thus $|S_1| = m$ and $|S_2| = n$

A mapping $\phi : S \rightarrow S^*$ from the original d dimensional input feature space S to a $(m + n)$ dimensional new feature space S^* is defined as follows.

1. Let $S_1 = \{\underline{s}_{1,1}, \underline{s}_{1,2}, \cdots, \underline{s}_{1,m}\}$ and $S_2 = \{\underline{s}_{2,1}, \underline{s}_{2,2}, \cdots, \underline{s}_{2,n}\}$

2. Let $\underline{x}_i \in S$, be a point in the input feature space. The transformed point $\phi(\underline{x}_i) \in S^*$. Let $\phi(\underline{x}_i) = [\Phi_{i,1}, \Phi_{i,2}, \cdots, \Phi_{i,m+n}]$ where $\Phi_{i,j}$ represents the value of the jth feature $(j = 1, 2, \cdots, (m + n))$ in $\phi(x_i)$.

3. $\phi(\underline{x}_i)$ is then calculated as follows

$$\Phi_{i,j} = d(\underline{x}_i, \underline{s}_{1,j}),\ j = 1, 2, \cdots, m \tag{3.1}$$

$$\Phi_{i,m+k} = d(\underline{x}_i, \underline{s}_{2,k}),\ k = 1, 2, \cdots, n \tag{3.2}$$

where $d(\cdot, \cdot)$ is the distance between the points, and set to euclidean metric in our implementation.

4. The class label of $\phi(\underline{x}_i)$ is set to θ_i, the class label for $\underline{x}_i$.

5. Steps 2, 3 and 4 are repeated for all values of $i = 1, 2, \cdots, N$.

The resultant feature space S^* is of dimensionality N, where $N = m + n$, and there are a total of N samples in the transformed training set.

3.3 Data Condensation

The proposed method of feature transformation calculates pairwise distance from a training point $x \in \mathbf{R}^d$ to every other point in the training set and this forms the new feature vector $z \in \mathbf{R}^N$. With increase in the number of training points the dimension z increases, until it may so happen that $N \gg d$. Therefore it is necessary to carry out some form of data condensation by retaining only the *representative points* of a class in the reduced training set, prior to the feature transformation described above.

The data condensation algorithm used in this implementation was originally proposed by Mitra, Murthy and Pal [17]. The algorithm proceeds as follows:

Let $S = \{\underline{x}_1, \underline{x}_2, \cdots \underline{x}_N\} \subseteq \mathbf{R}^d$ be the set of training points in the input feature space.

Let K be a positive integer.

Let $A = \Phi$

1. Let r_i be the minimum value of radius for which the number of points in a disc of radius r_i with centre as $\underline{x}_i$ contains K points, $i = 1, 2, \cdots, N$. r_i is the distance (euclidean) of the Kth nearest point from x_i.

2. Let $\underline{x}_{i0}$ be the point having radius $r_{i0} = \min\limits_{i=1,\ldots,N}\{r_i\}$.

3. Set $A = A \cup \{\underline{x}_{i0}\}$

4. Let $B_{i0} = \{\underline{x} \in S : d(\underline{x}, \underline{x}_{i0}) \leq r_{i0}\}$

5. Set $S = S - B_{i0}$

6. Repeat Steps 1 through 5 while $r_{i0} \leq r_{th}$ where r_{th} is a threshold on the minimum radius.

At the end of the procedure the set A contains the reduced training dataset.

3.4 Classification framework

Now that the features corresponding to each ROI image is available, the task is to perform binary classification. The most commonly used classifier for human detection is SVM. SVM as a baseline classifier has the advantages of being the one of the most efficient, reliable and scalable classifiers. The following properties of linear SVM make it popular - it converges reliably during training, it is scalable, i.e, can handle large data sets, and has good robustness towards different choices of feature sets and parameters.

Dalal and Triggs [6] use Linear SVM (soft margin C = 0.01) as a baseline classifier on HOG features, hence we use the same on the transformed feature space in order to make a direct comparison. Linear SVM ensures that the feature set is as linearly separable as possible in the input feature space, so improvements in performance imply an improved encoding of features from the input image. Greater the separation of the points in the two classes, lesser is the number of misclassifications by the linear SVM.

First a linear SVM is trained using a set of positive (human) and negative(non-human) training images. The classifier is then used to classify each test image ROI. The confidence of the SVM detection corresponds to the distance of the feature point from the hyperplane of separation.

Chapter 4

Experimental Results for Classification

The performance of classification using the proposed feature transformation is evaluated against the usage of the default HOG feature space, with the same classifier configuration. The training and testing datasets used for evaluation consist of images, since the authors who proposed HOG [6] had applied their method for human detection in images, and the image dataset used in their implementation, the INRIA Person dataset [1], has become the benchmark for evaluation of all human detection implementations.

4.1 Image Dataset Description

The full training dataset consists of 2416 positive images in the INRIA Person training dataset. The negative training samples are 1542 images of cars. Since the video surveillance datasets used for tracking consist of cars as the only moving ojects, we have manually selected images of cars from various Car datasets. The testing data consist of 1218 images of persons in the INRIA test dataset, and 854 images of cars. Figure 4.1 shows some of the training samples in our training dataset.

4.2 Classifier configuration

A soft-margin ($C = 0.01$) SVM with a linear kernel has been used as baseline classifier in both the state of the art and the proposed method for human detection.

(a) (b) (c) (d)

Figure 4.1: (a) and (b) are samples from the INRIA Person dataset, (c) and (d) are images from the car dataset

4.3 Dataset configuration

The human detection method has been implemented under two configurations of the training dataset:

- The full training set consisting of 2416 positive and 1542 negative examples is used for feature extraction followed by classification.

  The default HOG feature vector has dimension 3780.

  The feature transformation proposed in Section 3.2 result in 3958 dimensional feature vector.

- The data condensation algorithm described in Section 3.3 is first applied to reduce the number of training points and then feature extraction is performed. The parameters of the condensation algorithm have been experimentally determined as follows:

  1. $K = 2$
  2. $r_{th} = 7$ for the positive training set, and $r_{th} = 8$ for the negative training set.

  The reduced training set consists of 643 positive examples and 433 negative examples. The straightaway consequence of the data condensation procedure is the reduced dimension of the features computed by our proposed method.

  The new feature size is 1076, while the original feature size for HOG remains 3780.

Under both configurations, the comparison of the proposed method is done with the state of the art.

4.4 Evaluation metrics

The metrics used for performance evaluation are:

- **False Positive count** *(FP)*

  The number of non-human test images incorrectly classified as human.

- **False Negative count** *(FN)*

  The number of test images of persons incorrectly classified as non-human.

- **Misclassification count**

  (FP + FN)

- **Recall/Hit Rate**

  $\frac{TP}{(TP+FN)}$, where TP is the True positive count.

- **Precision**

  $\frac{TP}{(TP+FP)}$

- **Accuracy**

  $\frac{(TP+TN)}{(TP+FP+TN+FN)}$, where TN is the True negative count.

4.5 Results

The following table presents the results of the human detection algorithm for both the existing and proposed implementations.

Method	Training Data Size	FN	FP	FP+FN	Recall	Precision	Accuracy
Proposed Features + SVM	Full (2416 pos + 1542 neg)	42	58	100	0.965	0.953	0.9517
	Reduced (643 pos + 433 neg)	32	72	104	0.9737	0.9428	0.9498
HOG Features + SVM	Full (2416 pos + 1542 neg)	54	69	123	0.9557	0.944	0.9406
	Reduced (643 pos + 433 neg)	58	80	138	0.9523	0.9354	0.9307

Table 4.1: Classification Results on INRIA dataset

4.6 Discussion

Our proposed method gives higher confidence of classification when compared to the classification using the state of the art HOG features. This is because the separation of points in the feature space is higher, lead to less number of misclassifications. Furthermore, if we perform data condensation prior to feature transformation, the dimensionality reduced to 1076 compared to 3780 for standard HOG. With a much lower dimension, the accuracy is better than the state of the art, as depicted in Table 4.1.

However, an anomaly was observed during the experiments. The HOG feature constructs the histogram bins using gradient orientations, and votes into the histogram based on gradient magnitude. If an image region contains constant intensity, the gradient magnitude is zero along both the x and y axis, and the gradient orientation is undefined over that region. If the region fully occupies a cell/block, that cell/block will contain a value of zero in all the histogram bins (since there is no histogram bin for the gradient orientation when it is undefined). In a normalized histogram, the sum of the bins should be 1, but in this case the sum will be zero. Euclidean distance between corresponding histograms of two points will give a low value of distance because of the zeros, and the ultimately there will be inaccurate representation of the object in the transformed feature space, giving rise to classification error.

Chapter 5

Tracking and Counting

Detecting humans among the moving objects in a video by binary classification was the focus of the previous chapter. This chapter discusses the problem of associating each such detected person with a motion trajectory, i.e, a path followed by that particular person in the video frames over time. A motion trajectory indicates the spatial location, size and motion information of the detected person in the previous frames as well as the predicted position in the subsequent frame. In order to achieve this it is necessary to define a state model, representing the history of spatial coordinates and motion information, and an observation model, representing the actual observed features in the current frames. Using the state model and the observation model, a statistical framework can be used predict the next state from the previous states with the help of actual observations. Therefore, a tracking system identifies the position and motion information for the detected person on the basis of the state information in previous frames, and the most recent observed features.

5.1 Particle Filters

A particle filter is a Bayesian filter, which is used for probabilistic state estimation in state-evolving tasks. In object tracking, the state to be estimated is the configuration of the trajectory associated with a detected object. A particle filter can be used to track a single object, given a state model and an observation model. Tracking multiple objects requires a method of data association in addition to the single object state estimation by the particle filter.

A particle filter estimates the conditional probability density function (pdf) of the state variables. It approximates samples from the state posterior distribution with a set of particles, in the presence of noisy or partial observations. Unlike the

Kalman filter, the state-space model can be nonlinear and the initial state and noise distributions need not follow gaussian distribution. In the case of human tracking, where the motion of a person is mostly non-linear, and there is noise in the form of missing detections from the previous stage, it is advantageous to use a particle filter for motion path estimation. The Sequential Importance Resampling (SIR) filters, also known as the bootstrap particle filter, has been used for person tracking. The Condensation framework [14] describes the bootstrap particle filtering technique, that forms the basic skeleton of our tracking implementation.

5.2 Single Person Tracking Framework

The tracking framework follows the conditional density estimation in [14], with our proposed observation model.

5.2.1 *Bootstrap Particle Filtering*

Let the state of the object at time t be denoted by x_t. The history of observations(features) at time t is denoted by $Z_t = \{z_1, z_2, \cdots, z_t\}$, and state history at time t is $X_t = \{x_1, x_2, \cdots, x_t\}$. A basic assumption is that the object state form a temporal Markov chain.

$$p(x_t|X_t) = p(x_t|x_{t-1}) \tag{5.1}$$

The objective is to estimate the state conditional density $p(x_t|Z_t)$ with the help of the state history and recent observations. The equation governing the propagation of the state density of the object over time is derived from Bayes decision rule.

$$p(x_t|Z_t) = k_t p(z_t|x_t) p(x_t|Z_{t-1}) \tag{5.2}$$

where

$$p(x_t|Z_{t-1}) = \int_{x_{t-1}} p(x_t|x_{t-1}) p(x_{t-1}|Z_{t-1}) \tag{5.3}$$

and k_t is a normalization constant that does not depend on x_t.

The state conditional density $p(x_t|Z_t)$ is approximated by a set of samples $\{s_t^{(n)} : n = 1, 2, \cdots, N\}$, also known as particles, randomly sampled from the distribution, having weights $\pi_t^{(n)}$ which represent the sampling probability. The sample set is represented as $S_t = \{(s_t^{(n)}, \pi_t^{(n)}) : n = 1, 2, \cdots, N\}$.

At time t_0 the samples $\{s_0^{(1)}, s_0^{(2)}, \cdots, s_0^{(N)}\}$ are chosen from the prior density $p(x)$, with $\pi_0^{(n)} = \frac{1}{N}$.

At time t the new sample set S_t is constructed from the old sample set S_{t-1} as follows:

1. **Selection**

   Randomly select a sample (with replacement) $s_{t-1}^{(i)}$ with probability of selection $\pi_{t-1}^{(i)}$ and assign $\hat{s_t}^{(n)} = s_{t-1}^{(i)}$ for $n = 1, 2, \cdots, N$. Thus a new set of N particles is sampled from the old set of samples with a higher probability of sampling the particles with greater weight.

2. **Prediction**

   For each selected sample $\hat{s_t}^{(n)}$ generate a new sample $s_t^{(n)}$ by sampling from $p(x_t|x_{t-1}) = \hat{s_t}^{(n)}$. For instance, the following equation can be used to represent the dynamic state model

   $$s_t^{(n)} = F(\hat{s_t}^{(n)}) + \eta_t^{(n)} \tag{5.4}$$

   where function $F(\cdot)$ represents the deterministic drift and $\eta_t^{(n)}$ represents the stochastic diffusion (random noise) for the nth sample at time t.

3. **Correction**

   Update the weights of the new sample set by measuring the likelihood of observed features z_t with respect to the predicted state.

   $$\pi_t^{(n)} = p(z_t|x_t = s_t{}^{(n)}), n = 1, 2, \cdots, N \tag{5.5}$$

5.2.2 *Motion Model*

The dynamic state transition model predicts x_t from x_{t-1} based on a linear or non-linear model of dynamic drift and stochastic diffusion as in Eq. 5.4. This is also known as state propagation from x_{t-1} to x_t.

The state $X = \{x, y, v_x, v_y, s_x, s_y\}$ consists of the 2D object coordinates (x, y) in the video frame, its velocity (v_x, v_y) and scale (s_x, s_y) along the x and y directions respectively. It is assumed that acceleration is constant along both x and y axis.

The state transition equations used in this implementation are similar to [4].

$$(x, y)_t = (x, y)_{t-1} + (v_x, v_y)_{t-1}\Delta t + N(0, \sigma^2) \tag{5.6}$$

$$(s_x, s_y)_t = (s_x, s_y)_{t-1} + N(0, \sigma_s^2) \tag{5.7}$$

where σ^2, is the variance of the noise for the position and velocity variables, which follow Gaussian distribution with zero mean. Similarly the noise for the scaling variables are assumed to follow Gaussian distribution with zero mean and variance σ_s^2. The next state is thus predicted from previous state using the motion model.

5.2.3 *Proposed Observation Model*

The conditional probability density of the observations given the propagated state is needed to estimate the state conditional density as in Eq. 5.2. The likelihood of the observation z_t given the propagated state x_t of the nth particle is estimated by it's weight $\pi_t^{(n)}$ (Eq. 5.5). Thus each stage of state propagation using the Motion model is followed by a particle weight update stage in order to correct the prediction on the basis of actual observations. The weight update function gives more weight to the particles whose actual observed features are more closely associated with the predicted object state.

In our human tracking model, the input to the tracker is the bounding rectangle of the detected person. The state $X = \{x, y, v_x, v_y, s_x, s_y\}$ represents the coordinates of the centre of the bounding rectangle, its velocity and the scale along x and y axis respectively.

The predicted state X_t corresponds to a predicted location, velocity and scale of the bounding rectangle at time t. The actual position and dimensions of the bounding rectangle is the also available from the previous classification stage. The HOG feature vector z_{pred} is computed for the predicted bounding rectangle coordinates and scale. The HOG feature vector z_{obs} for actually observed detection rectangle is also computed. Then the Observation model equation for updating weight of the nth particle at time t is as follows:

$$\pi_t^{(n)} = p(z_t|x_t^n) = e^{-d^2(z_{pred}, z_{obs})} \tag{5.8}$$

where $d^2(\cdot, \cdot)$ is the square of the euclidean distance function.

5.3 Multiple Persons Tracking framework

The single person tracking framework is extended to multi-person tracking by using a particle filter for each detected person, and a data association method for assigning detections to tracks. Since there are multiple tracks, additional track management techniques are also required.

5.3.1 *Track Initialization*

A track is initialized when there is a detected person who is not yet assigned to any of the existing tracks. In our implementation, a track is initialized only if the detection confidence exceeds a minimum threshold.

5.3.2 *Track Association*

When there are multiple detected persons in the same video frame, and there are multiple tracks corresponding to previous detections, a set of association rules are defined to associate detections with tracks.

1. At most one detection is assigned to at most one track.
2. If the number of detections exceeds the number of tracks, a new track is created for the unassigned detections, as described under *Track Initialization*.
3. If the number of tracks exceeds the number of detections, then the remaining tracks which are not assigned any detection are updated to reflect the total number of missed detections in latest consecutive frames.
4. If the number of tracks and detections are equal
   (a) A track is assigned the detection whose coordinates lies within a radius with centre at the latest recorded track coordinates.
   (b) Once the detection is assigned to a track, it cannot be assigned to another track even though it falls within the radius of another track. Thus the method of assigning detections to tracks is a Greedy approach.
   (c) If there are more than one detections that fall within the radius of the track, then the HOG features of all the detections are computed and the nearest neighbour from the recorded track HOG is assigned to the track.

5.3.3 *Track Update*

The particle filters for tracks which are assigned a detection are updated by the calculating the new particle weights on the basis of the observed detections, and re-sampling to select new particles with the largest weight.

The tracks which are not assigned any detection are updated to reflect their new *missed detection count*, i.e, the latest number of consecutive frames in which the track is not assigned any detection.

5.3.4 *Track Deletion*

The obsolete tracks are deleted if any of the following conditions hold:

1. If the age of the track exceeds a threshold and throughout the track lifetime there have been too few assigned detections.

2. If the *missed detection count* of a track exceeds a predefined threshold.

3. If the track is not a young one, and the bounding rectangle coordinates lie very close to the frame boundary and there are consecutive missed detections for the last few frames, then it is assumed that the person who was being tracked has walked out of the video frame, and the track is deleted.

Chapter 6

Experimental Results for Tracking and People Counting

The count of distinct people in a video is equal to the number of trajectories initiated in the Tracking phase. The performance of our human detection and tracking implementation has been evaluated with the help of test video taken from publicly available surveillance video datasets.

6.1 Training Dataset Description

The training data used is the full image training dataset described under Section 4.1. This is used to train a linear SVM classifier (Section 4.2). The trained classifier is tested on cropped image ROIs from the frames of a test video.

6.2 Testing Dataset Description

Six video sequences used for testing have been taken from the PETS 2000, 2001 and 2009 [2] surveillance datasets. These fixed camera videos depict outdoor scenes, and moving objects consist of cars and people. The people are upright, with frontal, back, or side view of the full body. In all the test videos at any point of time there are not more than two or three people in close vicinity in order to avoid the complexities of data association in a multi-tracking framework. Occlusions of people are only of short duration. In each complete video there are up to four or five walking people and one or two moving cars.

6.3 Implementation details

The person detection and tracking system was implemented in C++ using OpenCV library for video processing utility functions. The training and testing was performed on a system with Intel Core i5 (2.50 GHz) processor and 4 GB RAM.

6.4 Evaluation metrics

The metrics used for evaluation are same as in Section 4.4. Only this time False Positive refers to a tracking of an object that is not human, and False Negative refers to a human that is not tracked in the video.

6.5 Results

The following table presents the results of tracking using the default HOG features as well as the proposed features for classification.

Video Dataset	**Features for classification**	**Avg Precision**	**Avg Recall**	**Avg Accuracy**
PETS 2009(S0)	HOG	0.735	0.835	0.75
	Proposed Features	1.0	1.0	1.0
PETS 2000	HOG	0.585	0.585	0.515
	Proposed Features	1.0	0.875	0.9
PETS 2001	HOG	0.8	1.0	0.835
	Proposed Features	1.0	1.0	1.0

Table 6.1: Results of Human Detection and Tracking

The results of human detection and tracking using the proposed method are shown in Figure 6.1.

6.6 Discussion

The background subtraction algorithm is based on pixel intensity values. Shadows are often detected as foreground objects and are sent to the classification stage. Colour match of person's garments with the background creates multiple disconnected components for a single person, each component separately processed for classification and tracking. Since the classifier performs binary classification with

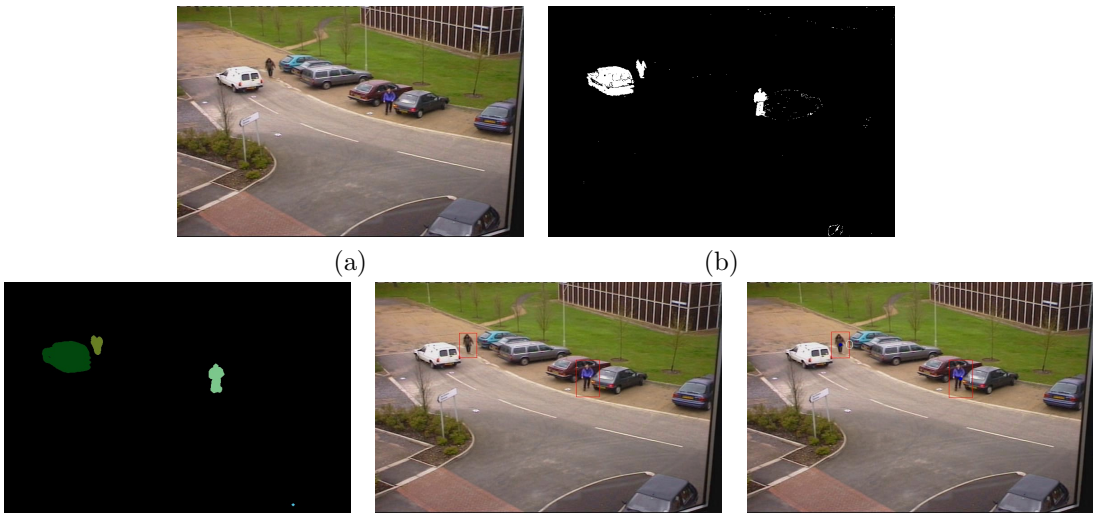


Figure 6.1: (a) is the original video frame, (b) and (c) are the results after background subtraction and connected component analysis respectively, (d) is the classifier result where a bounding box denotes the human class, (e) is the tracking result

images of persons and cars only as training data, such anomalies in background subtraction lead to misclassifications, and false track initializations.

To handle the above problems to a certain extent, a track is initialized only if there are high confidence detection results for three consecutive frames. Since false detections due to changes in background do not persist over time, this helps reduce most of the false track initializations. Since our proposed feature space gave higher confidence of detections than using HOG features, false track initializations could be prevented with a high threshold, whereas it was not possible to define such a threshold for the existing method.

The following challenges identified during the experiments, remain yet to be solved.

- If a two or three people move together throughout the time they are observed in the video, there is a single detection and a single trajectory associated with all the people, and hence they are counted as a single person.
- If a moving person is occluded by another object for a large number of frames then the previous trajectory terminates and a new trajectory is initiated for the person when he/she is again detected.
- If two persons cross over their trajectories labels may get interchanged, but the total count remains the same.

Chapter 7

Conclusion

This thesis targeted to solve the human detection and tracking problem with a focus on how to increase classification accuracy and minimize the number of false detections. A method for obtaining new features based on pairwise distance between existing features was proposed, and the method showed improved performance over the state of the art HOG feature space used for human classification. In order to make the proposed feature extraction method scalable with the number of training samples, a data condensation technique for training set reduction was used prior to feature extraction, and this further resulted in dimensionality reduction of the proposed feature space. Even with training set reduction, the proposed method achieved higher rate of precision, recall and accuracy than the state of the art on benchmark image dataset.

The detection algorithm was followed by a tracking mechanism using particle filters for state prediction using a proposed observation model. Data association rules and conditions for track management have been defined for a multiple object tracking framework. Performance comparison of tracking using proposed features against tracking using existing HOG features reveals a higher accuracy of the the proposed method on several test video datasets.

The proposed method for feature extraction has been carried out on HOG features only. Nowadays many implementations use a combination of other features with HOG for human detection, evaluation of the proposed method on those features can be a future scope for improvement in classification performance. Also the some of the issues surrounding multi-object tracking could not be solved in the current implementation. Robust tracking in the presence of missing detections, and false positives were partly handled in this thesis. Data association rules for many challenging cases are yet to be formulated, this leaves future scope for improved efficiency in tracking and people counting.

Bibliography

[1] http://pascal.inrialpes.fr/data/human/.

[2] http://www.cvg.reading.ac.uk/slides/pets.html.

[3] Yaakov Bar-Shalom. *Tracking and data association*. Academic Press Professional, Inc., 1987.

[4] Michael D Breitenstein, Fabian Reichlin, Bastian Leibe, Esther Koller-Meier, and Luc Van Gool. Robust tracking-by-detection using a detector confidence particle filter. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 1515–1522. IEEE, 2009.

[5] Michael D Breitenstein, Fabian Reichlin, Bastian Leibe, Esther Koller-Meier, and Luc Van Gool. Online multiperson tracking-by-detection from a single, uncalibrated camera. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(9):1820–1833, 2011.

[6] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE, 2005.

[7] Navneet Dalal, Bill Triggs, and Cordelia Schmid. Human detection using oriented histograms of flow and appearance. In *Computer Vision–ECCV 2006*, pages 428–441. Springer, 2006.

[8] Ahmed Elgammal, David Harwood, and Larry Davis. Non-parametric model for background subtraction. In *Computer Vision—ECCV 2000*, pages 751–767. Springer, 2000.

[9] William T Freeman and Michal Roth. Orientation histograms for hand gesture recognition. In *International workshop on automatic face and gesture recognition*, volume 12, pages 296–301, 1995.

[10] Dariu M Gavrila. Pedestrian detection from a moving vehicle. In *Computer Vision—ECCV 2000*, pages 37–49. Springer, 2000.

[11] Dariu M Gavrila, Jan Giebel, and Stefan Munder. Vision-based pedestrian detection: The protector system. In *Intelligent Vehicles Symposium, 2004 IEEE*, pages 13–18. IEEE, 2004.

[12] Dariu M Gavrila and Stefan Munder. Multi-cue pedestrian detection and tracking from a moving vehicle. *International journal of computer vision*, 73(1):41–59, 2007.

[13] Rob Hess and Alan Fern. Discriminatively trained particle filters for complex multi-object tracking. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 240–247. IEEE, 2009.

[14] Michael Isard and Andrew Blake. Condensation—conditional density propagation for visual tracking. *International journal of computer vision*, 29(1):5–28, 1998.

[15] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[16] Wei-Lwun Lu, Kenji Okuma, and James J Little. Tracking and recognizing actions of multiple hockey players using the boosted particle filter. *Image and Vision Computing*, 27(1):189–205, 2009.

[17] Pabitra Mitra, CA Murthy, and Sankar K Pal. Density-based multiscale data condensation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(6):734–747, 2002.

[18] Timo Ojala, Matti Pietikäinen, and Topi Mäenpää. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(7):971–987, 2002.

[19] Kenji Okuma, Ali Taleghani, Nando De Freitas, James J Little, and David G Lowe. A boosted particle filter: Multitarget detection and tracking. In *Computer Vision-ECCV 2004*, pages 28–39. Springer, 2004.

[20] Constantine Papageorgiou and Tomaso Poggio. A trainable system for object detection. *International Journal of Computer Vision*, 38(1):15–33, 2000.

[21] Donald B Reid. An algorithm for tracking multiple targets. *Automatic Control, IEEE Transactions on*, 24(6):843–854, 1979.

[22] Payam Sabzmeydani and Greg Mori. Detecting pedestrians by learning shapelet features. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007.

[23] Adrian Smith, Arnaud Doucet, Nando de Freitas, and Neil Gordon. *Sequential Monte Carlo methods in practice*. Springer Science & Business Media, 2013.

[24] Chris Stauffer and W Eric L Grimson. Adaptive background mixture models for real-time tracking. In *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, volume 2. IEEE, 1999.

[25] Paul Viola, Michael J Jones, and Daniel Snow. Detecting pedestrians using patterns of motion and appearance. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 734–741. IEEE, 2003.

[26] Xiaoyu Wang, Tony X Han, and Shuicheng Yan. An hog-lbp human detector with partial occlusion handling. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 32–39. IEEE, 2009.

[27] Christopher Richard Wren, Ali Azarbayejani, Trevor Darrell, and Alex Paul Pentland. Pfinder: Real-time tracking of the human body. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 19(7):780–785, 1997.

[28] Bo Wu and Ram Nevatia. Detection of multiple, partially occluded humans in a single image by bayesian combination of edgelet part detectors. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 1, pages 90–97. IEEE, 2005.

[29] Bo Yang and Ram Nevatia. Online learned discriminative part-based appearance models for multi-human tracking. In *Computer Vision–ECCV 2012*, pages 484–498. Springer, 2012.

[30] Changjiang Yang, Ramani Duraiswami, and Larry Davis. Fast multiple object tracking via a hierarchical particle filter. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 1, pages 212–219. IEEE, 2005.

[31] Qian Yu, Gérard Medioni, and Isaac Cohen. Multiple target tracking using spatio-temporal markov chain monte carlo data association. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007.

[32] Min-Ling Zhang and Lei Wu. Lift: Multi-label learning with label-specific features. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 37(1):107–120, 2015.

[33] Qiang Zhu, Mei-Chen Yeh, Kwang-Ting Cheng, and Shai Avidan. Fast human detection using a cascade of histograms of oriented gradients. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 1491–1498. IEEE, 2006.