A Novel Pedestrian Detection and Tracking with Boosted HOG Classifiers and Kalman Filter

Penny Chong  
Lee Kong Chian Faculty of Engineering and Science  
Universiti Tunku Abdul Rahman  
Selangor, Malaysia  
pennychong94@1utar.my

Yong Haur Tay  
Lee Kong Chian Faculty of Engineering and Science  
Universiti Tunku Abdul Rahman  
Selangor, Malaysia  
tayyh@utar.edu.my

Abstract—This paper focuses on developing a stable pedestrian detection and tracking algorithm. Although Histogram of Oriented Gradients (HOG) features are the best representation for human shapes, computing these feature vectors are computationally expensive as it slows down the overall detection process. Hence with the use of cascade of boosted classifiers, the overall process was shortened significantly even in the absence of graphics processing unit (GPU). Along with Kalman filter approach, the algorithm achieved desirable results in tracking pedestrians coming from various directions. The Kalman filter model with its self-correcting mechanism, guarantees that the tracking improves overtime as more raw detections are supplied. As long as consistent detections were supplied to the filter in the early stages, the tracking continues even when the detector becomes faulty.

Keywords—boosted classifiers; HOG; Kalman filter; pedestrian detection; pedestrian tracking

I. INTRODUCTION

Computer or machine vision is a field in artificial intelligence that has become great area of research due to its wide range of applications. From automated navigation of vehicles, counting applications to medical imaging, all these applications require the computer to process, understand and analyse a scenario to make intelligent decisions. With a large increase in the number of CCTVs, it has become impractical for human operators to monitor the video footage. This is where automated pedestrian detection and tracking system comes in handy.

With the advancement in computers and the development in this area, there are now various methods/algorithm to perform automated pedestrian detection and tracking. However, developing a reliable detection and tracking algorithm that can perform equivalently well in a crowded environment is still an issue. In a crowded environment where the degree of occlusion is high, it is often difficult for a computer to identify and determine the pedestrians’ paths.

The task becomes more complicated due to the variation of heights and body shapes of pedestrians. In an extreme case where two pedestrians have the same height, body shape and are wearing similar outfits, it may be quite difficult for a computer to differentiate and distinguish their paths. Illumination changes in the scene may also alter the appearance of the object causing it to look different. Therefore it is not easy for one to develop a reliable real-time algorithm to address all these issues. There is still no promising algorithm in terms of accuracy and speed.

In this problem, we have employed a HOG pedestrian detection as a feature representation of the human shape. For tracking, we used the Kalman filter approach together with the Hungarian algorithm. We have conducted two different experiments to compare the default person detector in OpenCV, an implementation of [3] with our trained boosted classifier. The final tracking results for both experiments were also compared with existing works [4], [5], [6]. The standard Town Centre video and the hand annotated ground truth data provided by the University of Oxford were used in the comparisons.

II. RELATED WORKS

A. Object Detection Methods

Generally there are three different types of approach used in object detection. They are the feature-based approach, motion-based approach and classifier-based approach [7].

In a feature-based approach, features such as colors and shapes are extracted as a representation of the image. Tian and his team [8] had employed a HOG feature descriptor and a Local Color Self Similarity Feature (LCSSF) in pedestrian detection. The detection time was shortened as compared to using a Color Self Similarity (CSS) feature.

In a motion-based approach such as background subtraction, the objects are detected based on movements. Rakibe and Patil [9] had employed background subtraction to detect humans in video frames. The moving humans were detected by finding the difference between the current frame and the background.

In a classifier-based approach, the classifier is trained to recognize or detect objects by feeding in positive and negative training samples. Dalal and Triggs [3] had used a Support Vector Machine (SVM) classifier with HOG features to classify human and non-human based on the optimal hyperplane. On the other hand, Viola and Jones [10] had proposed to use a cascade of
boosted classifiers for face detection where each layer was trained with the AdaBoost method.

B. Object Tracking Methods

Generally there are three different types of approach used in object tracking. Point tracking approach, kernel-based tracking approach and silhouette-based tracking approach [11], [12].

In point tracking approach, we represent the detected objects as points across frames [11], [12]. Jiang and his team [13] had employed a colour model together with a Kalman filter motion model in pedestrian tracking. Initially, a HOG-SVM classifier was used to detect the pedestrians. After the detection phase, Kalman filter was used to track the pedestrians.

In kernel tracking, a moving object is computed and represented by an embryonic object region from frame-to-frame and Kalman filter model in pedestrian tracking. Initially, a HOG-SVM classifier was used to detect the pedestrians. After the detection phase, Kalman filter was used to track the pedestrians.

For a silhouette-based tracking which is suitable for objects with complex shapes, colour histogram, contour or edges are commonly used [11], [12]. For instance, Sato and Aggarwal as cited in [14] had employed a silhouette matching technique using a Hough transform to calculate the trajectory of the moving object.

III. METHODOLOGY

The algorithm is divided into two phases. The detection phase is followed by the tracking phase. The video frames are input into the system on a frame-by-frame basis. In the feature extraction process, we compute the HOG feature descriptor. The computed feature vectors are fed into a binary classifier, either a SVM classifier (experiment 1) or a cascade of boosted classifiers (experiment 2).

Each detected pedestrian is assigned a unique identity. In the tracking phase, we employ the Hungarian algorithm and the Kalman filter model to estimate the locations of the pedestrians based on the raw detection results. The detection performance for cascade of boosted classifiers and tracking performances for both experiments were evaluated using receiver operating characteristic (ROC) curves [10] and CLEAR MOT metrics respectively [15].

A. Histogram of Oriented Gradients

We have employed the Histogram of Oriented Gradients (HOG) feature descriptor as a feature representation for the human shape as it is known to be invariant to photometric and geometric transformation [3]. Initially, the gamma values and colors were normalized. The gradient values were computed vertically and horizontally using the one-dimensional centered, point discrete derivative mask. Based on the gradient values, the pixels cast weighted votes into orientation cells. Finally, the gradient strengths are contrast-normalized by grouping cells into large spatial connected blocks to adapt better to illumination changes.

B. Support Vector Machines

Support Vector Machine (SVM) algorithm finds an optimal hyperplane that maximizes the separation between the hyperplane and the points in space. In a binary linear problem, the SVM algorithm groups the training samples into two different categories based on an optimal hyperplane that maximizes the distance between the two categories [16]. With the optimal hyperplane, the SVM classifier can classify new unseen examples into one of the category. Experiment 1 will use this approach.

C. Cascade of Boosted Classifiers

Cascade of boosted classifiers was proposed by Viola and Jones [10] for face detection. The cascade consists of many layers where each layer is a binary classifier trained with AdaBoost, a machine learning algorithm to classify images according to certain features. At any layer where a sub-window is rejected, there will be no further processing on that sub-window [10]. This speeds up the detection process since most of the sub-windows in a single image are negative sub-windows and can be eliminated quickly in the first few layers of the cascade. We employed this approach with HOG feature for pedestrian detection in experiment 2. We have trained two different detectors using Daimler [17] training samples and INRIA [3] training samples.
D. Kalman Filter Model

The Kalman filter model is a linear quadratic estimation (LQE) model that is used to estimate the state of an object in an on-going process. The Kalman filter model is able to give an estimation of the state by recursive computations through two main stages.

The first stage is the time update stage, also known as the prediction stage [1]. It is a stage where the filter will predict the current state using observations from the previous state. The second stage is the measurement update stage also commonly known as the correction stage [1], as this is a stage where the filter will refine its prediction and correct its estimation by taking into account the actual measurements. The two stages can be simplified into the following equations, where equation (1) represents the prediction stage and equation (2) represents the correction stage [1].

\[
\hat{x}_k = A\hat{x}_{k-1} + Bu_{k-1}, \quad (1) \\
\hat{x}_k = \hat{x}_k + K_k(z_k - H\hat{x}_k). \quad (2)
\]

The variables are defined as follows. \(\hat{x}_k\) is the priori estimate at time \(k\), \(\hat{x}_k\) is the posteriori estimate at time \(k\), \(u_{k-1}\) is the optional control input at time \(k-1\), \(z_k\) is the actual measurement/location at time \(k\), \(K_k\) is the Kalman gain at time \(k\) while \(A\), \(B\) and \(H\) are matrices with different dimensions [1]. From equation (1) and (2), we observed that the output from the prediction stage will be used as an input for the correction stage. Therefore the estimate improves overtime as more information is supplied.

In the case where the detection fails or the pedestrian is occluded, the filter can still predict the current location of the pedestrian based on previous observations. If the pedestrian is not detected for a consecutive number of frames, the pedestrian is assumed to have left the scene. In order to identify and track pedestrians, the filter works closely with Hungarian algorithm [2] which is a data association algorithm to find the correspondences between individuals and tracks.

IV. EXPERIMENTAL SETUP

For both experiments, the Town Centre full HD video provided by the University of Oxford [4] with its hand annotated ground truth were used to test the performance of our algorithm. The video is 3 minutes long, with 25 frames per second and an average of 16 pedestrians per frame. The algorithm is implemented using Open Source Computer Vision (OpenCV) 2.4.9 in C++ language. A comparison of our tracking results in experiment 1 and 2 with existing works [4], [5], [6] will be discussed in a later section.

A. Experiment 1

In the first experiment, we used the SVM classifier approach with HOG feature for pedestrian detection. We do not train our own classifier in this experiment. Instead, we used the HOG-SVM default people detector in OpenCV which is an implementation of [3]. The raw detections are passed on to our tracker. At the end of the 3 minutes video, the tracking output which contains information on the location of the pedestrians will be stored in a json format and be evaluated to measure our tracking performance.

B. Experiment 2

In the second experiment, we trained our own cascade of boosted classifiers [10] using the AdaBoost method to perform pedestrian detection. We trained two different HOG detectors using Daimler training samples [17] and INRIA training samples [3]. Daimler training samples consist of 15,660 positive samples and 6,744 negative samples [17]. On the other hand, INRIA training samples consist of 2,416 positive samples and 1,218 negative samples [3]. The Daimler and INRIA detectors were compared in terms of detection performance. The better detector in this experiment is chosen and used in the tracking phase. Similar to experiment 1, the tracking output is stored in a json format.

V. RESULTS AND FINDINGS

In experiment 2, we trained two different detectors based on different training samples. It was observed that the Daimler detector outperforms the INRIA detector when tested on Town Centre video [4].

![ROC Curve](image)

Fig.2. ROC curve for Daimler detector and INRIA detector.

From the ROC plot in Fig. 2, it can be observed that the detector fed with Daimler training samples [17] shows a lower false positive rate as compared to the detector fed with INRIA training samples [3], at a specific hit rate. Daimler detector reaches a hit rate of 90% much earlier and at a lower false positive rate as compared to the INRIA detector. This implies that the Daimler detector has a better discriminative ability and has outperformed the INRIA detector.

Considering that the Daimler detector was trained with more positive and negative samples as compared to INRIA
detector, we expect the Daimler detector to have a better detection performance. Only with large training samples, we can cover a variety of angles and possibilities, resulting in a more robust detector. Using the Daimler detector, we will compare our tracking results against experiment 1 and other existing works [4], [5], [6] as shown in Table 1.


<table>
<thead>
<tr>
<th>Algorithm</th>
<th>MOTP</th>
<th>MOTA</th>
<th>Detection (sec/frame)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Benfold &amp; Reid 4</td>
<td>77.08 %</td>
<td>66.31 %</td>
<td>1.2</td>
</tr>
<tr>
<td>Dehghan, et al. [5]</td>
<td>71.93 % (estimate)</td>
<td>75.59 % (estimate)</td>
<td>-</td>
</tr>
<tr>
<td>Izadinia, et al. [6]</td>
<td>71.60 % (estimate)</td>
<td>75.70 % (estimate)</td>
<td>-</td>
</tr>
<tr>
<td>Experiment 1</td>
<td>67.26 %</td>
<td>50.61 %</td>
<td>4.5</td>
</tr>
<tr>
<td>Experiment 2</td>
<td>69.38 %</td>
<td>67.07 %</td>
<td>0.76</td>
</tr>
</tbody>
</table>

From Table 1, the algorithm proposed by Benfold and Reid [4] achieved the highest MOTP among the five algorithms. With a Kanade-Lucas-Tomasi (KLT) approach and a Markov-Chain Monte-Carlo Data Association (MCMCDA) technique, they obtained a MOTP of 77.08%. On the contrary, Izadinia and his team [6] achieved the highest MOTA which is 75.70% by constraining pedestrian tracking by parts tracking. Dehghan and his team [5] obtained similar results to Izadinia’s team [6] by using a part-based human detector and a data association method called the Generalized Minimum Clique Graphs. However the MOTP and MOTA values as reported in [5] and [6], are estimates only as we could not obtain their tracking output to be used on our evaluation algorithm. On the other hand, the MOTP and MOTA values for [4] as reported in Table 1 are based on the same evaluation algorithm as our experiments. Hence for a fair comparison, we will focus mainly on comparing our experiments with [4] results.

Our experiment 2 which uses a cascade of boosted classifiers shows more promising results as compared to our experiment 1 which uses SVM classifier. Comparing the two experiments, our detection time was shortened from approximately 4.5 seconds/frame to 0.76 seconds/frame. It is because the layers in the cascade are designed for fast detection by eliminating a large number of negative sub-windows [10]. The multiple object tracking precision (MOTP) and multiple object tracking accuracy (MOTA) values reported for experiment 2 are also better than experiment 1. In addition, our experiment 2 had achieved a MOTA value of 67.07% which is about 1% higher with better detection time as compared to [4].

Referring to Table 2, our algorithm shows a lower number of false positives as compared to [4]. Our proposed algorithm has reduced the number of false positives by half. In conjunction, our algorithm also has a lower number of mismatches. However out of the 71,460 ground truths, our algorithm found only 53,545 correspondences between hypotheses and ground truths while [4] found a total of 58,272 correspondences. In other words, our algorithm suffers from 25% misses while [4] only suffers from 18% misses which is the downside of our algorithm. The high number of misses have affected our MOTP value. Using only a segment of the video provided by [4], we have conducted another study to understand the factors that affect our number of misses during tracking.

![Fig.3. Error analysis on the factors affecting the number of misses.](image)

From Fig. 3, occlusion is observed to be the main reason for the high number of misses in our algorithm. There is always a high degree of occlusion between pedestrians in a semi-crowded environment. Thus to improve our results, our algorithm must be able to handle occlusion.

VI. CONCLUSION AND FUTURE WORK

By supplying more training samples, we can account for a larger variation in the shapes and postures of the human body leading to a better detection performance. The choice of classifier such as a boosted cascade in experiment 2 had shortened the entire process and yet it gives better MOTP and MOTA values as compared to SVM classifier in experiment 1, when employed together with Kalman filter. Although our algorithm in experiment 2 shows promising results in terms of speed and MOTA value, our MOTP value still lags behind. This can be improved if we are able to handle occlusion.
In crowded environments, one should try part-based detection to reduce the impact of occlusion on tracking performance. When a part of the body is occluded, a part-based detector can still detect the partially occluded body but a full body detector will fail. For instance, a head detector can be employed to estimate the full body region in pedestrian tracking.

REFERENCES
