Real-Time Pedestrian Detection and Tracking

Anmol J Bhattad [1], **Sharukh S Shaikh** [1], **Sumam David S.** [1], **K. P. Anoop** [2] and **Venkat R Peddigari** [2]

1 Department of Electronics and Communication Engineering, National Institute of Technology Karnataka, Surathkal, Mangalore 575025, India. Email: sumam@ieee.org
2 Formerly with Texas Instruments (India) Pvt. Ltd., Bangalore, India.

ABSTRACT: Pedestrian detection is a key problem in computer vision, with several applications that have the potential to positively impact the quality of life. This paper describes a comprehensive combination of feature extraction methods for vision-based pedestrian detection and tracking in Intelligent Systems based on monocular vision. First, we detect the pedestrian using Integral Channel Features and AdaBoost classifier, which is implemented with Modified Soft Cascade to achieve robust thresholds. Later we track the pedestrian for the next few frames based on Lucas Kanade features. The experiment results show that the method can detect and track pedestrian ahead of vehicle in spite of different sizes and postures. The algorithms have been tested on Inria database for the detection system and Caltech and Daimler datasets for the detection and tracking system.

Keywords: *Pedestrian Detection, Pedestrian Tracking, Integral Channel Features, Lucas Kanade features*

INTRODUCTION

Detecting and tracking people is an important area of research, and machine vision is bound to play a key role. Its applications include traffic control, robotics, entertainment, surveillance, care for the elderly and disabled, and content-based indexing. Our study aims to address these questions by detecting and tracking the pedestrians in the video which is at the rate of around 30 frames/second captured with a monocular camera mounted on the car. Fig. 1. shows the block diagram of the detection and tracking system.

Many interesting pedestrian detection approaches have been proposed in the literature. The Histogram of Oriented Gradients (HOG) detector proposed by Dalal and Triggs (2005), is a combination of edge orientation histograms, shape contexts and Scale-invariant feature transform (SIFT) descriptors. Histogram of Optical Flow (HOF) introduced by Dalal N. et. al. (2006), is based on motion features. HOF, is not affected by movements in camera and background, which makes it robust. In Stefan W. et. al. (2010), HOF is used in combination with HOG, and the model was trained using SVM.

Unlike features such as Haar, HOG, HOF, Sabzmeydani and Mori (2007) uses mid-level features called Shapelets to train the classifier. These features focus on local regions of the image and are built from low–level gradient information.

A multi cue approach suggested in Wojek and Schiele (2008) combines various features like Haar Wavelets, Haar-like HOG features, Shapelets and Shape Context and shows that a multi-cue pedestrian detection approach performs better than using single feature. The combination of modified HOG and HOF in Stefan W. et. al. (2010) performs better than any individual feature.

Viola P. et.al. (2005) is an extension of the rectangle filters from Viola and Jones (2004) to the motion domain. It describes a pedestrian detection system that integrates image intensity information with motion information. ChnFtrs, Dollar P. et. al. (2009) coupled the integral channel features with soft cascade AdaBoost algorithm which outperformed existing methods for pedestrian detection. Multiple image channels are computed using linear and non-linear transformations of the input image, and then features such as local sums, histograms, and Haar features are computed to get integral channels.

The overview of state-of-art pedestrian detectors brings out the following facts. HOG is a very basic and effective feature, but bare HOG can't give the best results. Hence, a multi cue pedestrian detection is favorable. Mid-level features such as Shapelets are computationally expensive and hence unsuitable for real-time detector implementation. The need to calculate the Optical Flow channels makes the computation of HOF features very slow. Thus, HOG+HOF, Stefan W. et. al. (2010), is not good for real-time applications, inspite of having good accuracies. The integral channel features (ICF), Dollar P. et. al. (2009), obtained from color channels, gradient histograms etc. are computationally efficient, capture thorough information and offer faster detection. The ChnFtrs utilizing ICFs detector has high accuracy but, has a rate of 1.18 fps. But for real-time implementation we need faster detectors. Higher computational time requirement makes the non-linear SVM kernels unsuitable for real-time pedestrian detection. Though SVM is universally known for good accuracies, AdaBoost is quite comparable. AdaBoost along with soft cascade is much faster than linear SVM, at the cost of marginal decrease in the performance and hence, it is chosen as the basic classifier in our implementation.

PEDESTRIAN DETECTION

Given an input image I, channels can be computed using linear or non-linear transformation of I. We consider the following channels for our detector:

LUV Colour Channels

The Fig. 3 shows the LUV channels of the original image shown in Fig. 2. In case of 8-bit images the values of R, G and B are converted to the floating-point format, scaled to fit 0 to 1 range and then converted into LUV channels.

Gradient Magnitude

The Gradient Magnitude measures the strength and distribution of the gradients within an image. Let *I(i,j)* denote an m×n discrete signal, and $\delta I/\delta x$ and $\delta I/\delta y$ denote the discrete derivatives of I (typically 1D centred first differences are used). Gradient magnitude *M(i,j)* and orientation *O(i,j)* are given by

$$M(i,j) = \sqrt{\frac{\delta I(i,j)^2}{\delta x} + \frac{\delta I(i,j)^2}{\delta y}} \tag{1}$$

$$O(i,j) = \arctan\left(\frac{\frac{\delta I(i,j)}{\delta y}}{\frac{\delta I(i,j)}{\delta x}}\right) \tag{2}$$

Fig. 4 is the Gradient Magnitude of the original image shown in Fig. 2. Let the horizontal changes given by $\delta I/\delta x$ be denoted by G_x and the vertical changes given by $\delta I/\delta y$ be denoted by G_y. The actual gradient magnitude is given by

$$G = \sqrt{G_x^2 + G_y^2} \tag{3}$$

To avoid the computational burden of calculating the square root for each pixel we adopt an approximation given by

$$G = |G_x| + |G_y| \tag{4}$$

Gradient Orientation Channels

The Gradient Orientations *O(i,j),* Dollar P. et. al. (2009) are distributed in the range [0, π]. The range is divided into equal sized six bins and the orientation channels are computed as

$$G_\theta(i,j) = M(i,j).\mathbf{1}\,[O(i,j) \,\epsilon\, (\theta_1, \theta_2)] \tag{5}$$

where $G_\theta(i,j)$ is the gradient orientation channel within range $[\theta_1, \theta_2]$, *M(i,j)* the gradient magnitude and *O(i,j)* is the gradient angle, respectively, at *I(i,j).* Fig. 5 shows the gradient orientation channels along six orientations within six equal sized bins. The six bins for the images shown are [0, π/6], [π/6, π/3], [π/3, π/2], [π/2, 2π/3], [2π/3, 5π/6] and [5π/6, π]. The gradient orientations are computed over these bins and are used for obtaining integral channel features.

Gradients are computed over the channels using the discrete derivative mask d^2x = [1,-2, 1] and d^2y = [1,-2,1]′. These provide the second order gradients which have strong response near edges and other sharp transitions.

Using the above mentioned linear and non-linear transforms on the input image, we get different channels. Further we compute the features of each channel by taking the sum over randomly selected rectangular regions. These features are computed efficiently from the channels by using integral images. Such features are called as integral channel features.

Each feature has three parameters that are generated randomly. Given that we compute different channels of our detection Window, a feature is calculated on a random channel, over a rectangle whose dimensions and position is also random.

Soft Cascade AdaBoost Classifier

In AdaBoost, weak classifiers are cascaded to produce strong classifier, John Lu (2010). The weak classifiers are very simple and are computationally inexpensive. The weak classifiers used here are 2-depth decision tree. Bourdev & Brandt's (2005) thresholding technique after every weak classifier is better than thresholding at the end, helping us remove negative samples at an early stage. Given the large number of negative windows in an image, this feature called as soft cascade makes AdaBoost faster.

Modified Soft Cascade

In the proposed method, a modified version of soft cascade is suggested. Instead of thresholding at the end of every weak classifier, it is performed only after a few weak classifiers, separated by a regular interval *n*. Consider an AdaBoost classifier consisting of *T* weak classifiers. Let, $c_t(x)$ denote the confidence by the weak classifier *t* for a sample *x*. Then, the thresholding is shown by the following algorithm,

```
$d \leftarrow 0$
$for\ t = 1 \dots T$
        $d \leftarrow d + c_t(x)$
        $if\ \ t \,\%\, n == 0\ then$
                $if\ d < \ r_i\ return\ false$
$return\ true$
```

To determine the threshold, consider a set of positive test images *i* = 1, 2, 3,... We run it over the trained AdaBoost. Let $c_j(i)$ be the value of the j^{th} weak classifier for the i^{th} image, where j= n, 2n, 3n,…..N.
Let, $M_j = \sum_i c_j(i)$ and $\sigma_j = \sum_i (c_j(i) - M_j)^2$. Then, the threshold r_j is given by,

$$r_j = M_j - k\sigma_j \qquad (6)$$

To get the thresholds, test images are used instead of training images because, the AdaBoost will fit over the training data very well when compared to the test data, resulting in higher thresholds. Thus, the modified soft cascade helps us achieve speed like that of soft cascade and results in more robust thresholds.

Detection over full images

To implement the detector over the complete image, downscale the image from its original size and keep on placing them one upon the other. Scale down until, $\lfloor \mathrm{ImageHeight/Scale} \rfloor > 128$ and $\lfloor \mathrm{ImageWidth/Scale} \rfloor$>64.

Between the two consecutive levels of the pyramid the scale factor is chosen to be 1.2. And between the two consecutive windows the distance (window stride) is 8 pixels. New height and new width of the image are given by,

$$NewHeight = \lfloor OrigHeight/Scale \rfloor$$
$$NewWidth = \lfloor OrigWidth/Scale \rfloor \qquad (7)$$

Scanning-window style classification of image patches typically results in multiple responses around the target object. We circumvent this by removing any detector responses in the neighborhood of detections with the locally maximal confidence score. This technique is known as Non-Maximal Suppression (NMS), Rosten E. et. Al. (2006).

PEDESTRIAN TRACKING

The tracking phase assumes that the person has already been detected. Since, the displacement of the pedestrian is not huge in short interval of time (10-15 frames), the tracking system need not be robust. Hence, we consider the point and kernel tracking algorithms, Li X. et. al. (2013), that are not computationally intensive.

Lucas Kanade (LK) feature tracker

- Detection of the feature points: Shi-Tomasi feature detector, Shi and Tomasi (1994), is used to detect the feature points in the bonding box provided by the pedestrian detector.
- Pyramidal Implementation Feature Tracker: The pyramidal implementation, Bouguet J. (2001), of the LK optical flow is used, Dollar P. et. al. (2009), to evaluate the position of the feature points in the consecutive frames.
- Evaluate the final position of the bonding box: The position of the feature points in the consecutive frames is known. The outlier points which do not lie on the pedestrian and the points which move insignificantly are eliminated. By computing the velocities of the remaining points the shift in the bonding box is calculated.

RESULTS

The experiments were carried out on INRIA database [9] which comprises of 2,416 positive training samples of size 96x160 and 1,126 positive testing samples of size 70x134 and 1,218 negative training images and 453 negative testing images. The performance of the detector is evaluated by obtaining the Miss Rate Vs False Positive per Window (FPPW) plots. Miss Rate is the ratio of positive samples mispredicted to the total number of positive samples in the dataset. All the accuracies are measured at 10^{-3} FPPW.

For pre-processing, we applied a 3x3 Gaussian smoothening on the images. After multiple experiments, (Fig.6) we conclude that 12,000 features, 5-15 pixel rectangle size and AdaBoost with 1000 weak classifiers give the most promising results, with an accuracy of 93.82% (Fig. 7). A few snap shots are given in Fig. 8 and Fig. 9.

The tracking system assumes that the pedestrian has already been detected and the pedestrian detector gives the bonding box coordinates centering the pedestrian. The tracking system is then expected to track the pedestrian for next 10 to 15 consecutive frames i.e. around 0.3-0.5 seconds. We have run the algorithm on video clips from Caltech datasets, Dollar P. et.al. (2012), and Daimler, Enzweiler M. (2009). These video samples can be seen at http://goo.gl/SKV59Z.

CONCLUSIONS

Based on the literature survey on the state of the art detection systems ChnFtrs Filter detector which uses ICFs with Soft Cascade AdaBoost classifier is a milestone in real-time pedestrian detection. Integral Channel Features extracted from LUV, Gradient magnitude and Gradient Orientation channels are chosen for our system because they offer better speed and performance. AdaBoost classifier, with soft cascade feature is proven to be faster than other classifiers like SVM, NN, etc. The proposed, modified soft cascade helps us achieve speed similar to that of soft cascade and resulted in more robust thresholds. Once the pedestrian in the frame is detected, the detected pedestrian is tracked using Lucas Kanade feature based tracking. In future, we can build an optimized real-time VLSI design/ FPGA / microprocessor based implementation for pedestrian detection and tracking system.

ACKNOWLEDGEMENTS

The authors gratefully acknowledge Texas Instruments India Ltd, Bangalore, for their support to this work.

REFERENCES

[1] Bouguet, Jean-Yves. (2001), Pyramidal implementation of the affine lucas kanade feature tracker description of the algorithm, Intel Corporation.
[2] Bourdev, Lubomir, and Jonathan Brandt, (2005), Robust object detection via soft cascade, IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 236-243.
[3] Dalal Navneet, and Bill Triggs, (2005), Histograms of oriented gradients for human detection, IEEE Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 886-893.
[4] Dalal, Navneet, Bill Triggs, and Cordelia Schmid, (2006), Human detection using oriented histograms of flow and appearance, ECCV, pp. 428-441.
[5] Dollar, Piotr, Christian Wojek, Bernt Schiele, and Pietro Perona, (2012), Pedestrian detection: An evaluation of the state of the art, IEEE Transactions on Pattern Analysis and Machine Intelligence.
[6] Dollár, Piotr, Zhuowen Tu, Pietro Perona, and Serge Belongie, (2009), Integral Channel Features, BMVC, vol. 2, no. 4, p. 5.
[7] Enzweiler M. and D. M. Gavrila, (2009), Monocular Pedestrian Detection: Survey and Experiments, IEEE Trans. on Pattern Analysis and Machine Intelligence, vol.31, no.12, pp.2179-2195.
[8] John Lu, Z. Q, (2010), The elements of statistical learning: data mining, inference, and

prediction, Journal of the Royal Statistical Society: Series A (Statistics in Society) 173, no. 3, 693-694.

[9] Li, X., Hu, W., Shen, C., Zhang, Z., Dick, A., & Hengel, A. V. D. (2013), A survey of appearance models in visual object tracking, ACM Transactions on Intelligent Systems and Technology (TIST), 4(4), 58.

[10] Piotr's Image & Video Matlab Toolbox, http:// vision.ucsd.edu/~pdollar/toolbox/doc/index.html

[11] Rosten, E., & Drummond, T. (2006), Machine learning for high-speed corner detection, Computer Vision–ECCV, Springer Berlin Heidelberg, pp. 430-443.

[12] Sabzmeydani, Payam, and Greg Mori, (2007), Detecting pedestrians by learning shapelet features, IEEE Conference on Computer Vision and Pattern Recognition, pp. 1-8.

[13] Shi, Jianbo, and Carlo Tomasi, (1994), Good features to track, 1994 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 593-600.

[14] Viola, Paul, and Michael J. Jones, (2004), Robust real-time face detection, International journal of computer vision 57, no. 2: 137-154.

[15] Viola, Paul, Michael J. Jones, and Daniel Snow, (2005), Detecting pedestrians using patterns of motion and appearance, International Journal of Computer Vision63, no. 2: 153-161.

[16] Walk, Stefan, Nikodem Majer, Konrad Schindler, and Bernt Schiele. (2010), New features and insights for pedestrian detection, IEEE Conf. Computer Vision and Pattern Recognition, pp. 1030-1037.

[17] Wojek, Christian, and Bernt Schiele, (2008), A performance evaluation of single and multi-feature people detection, In Pattern Recognition, pp. 82-91. Springer Berlin Heidelberg.

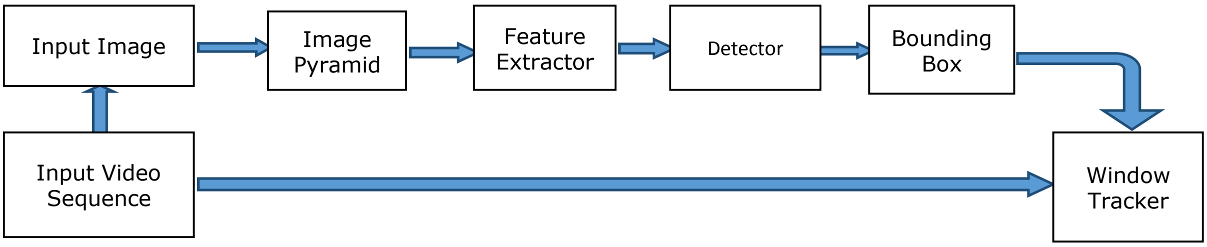


Fig.1 Block Diagram of the detection and tracking system

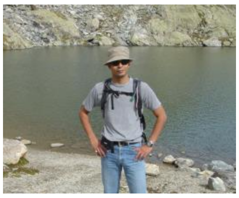


Fig. 2 Original Image

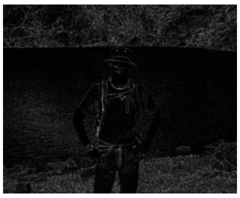


Fig. 3 Gradient Magnitude

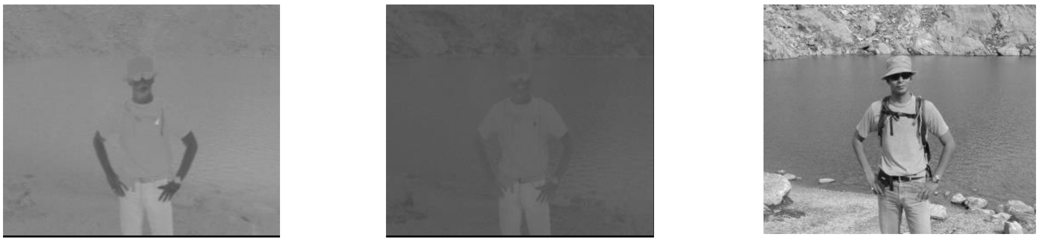


Fig. 4 CIE LUV Channels

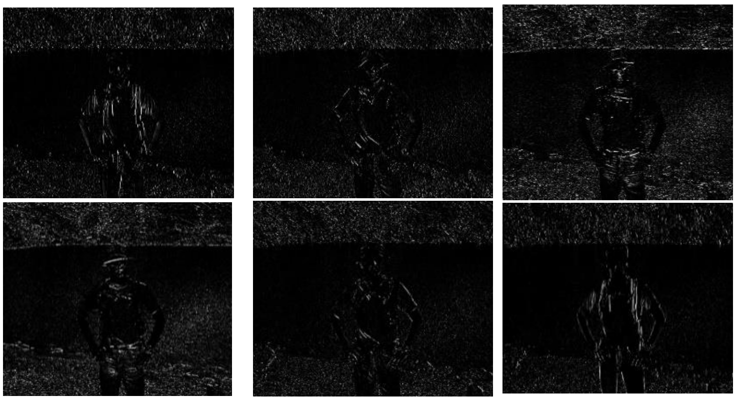
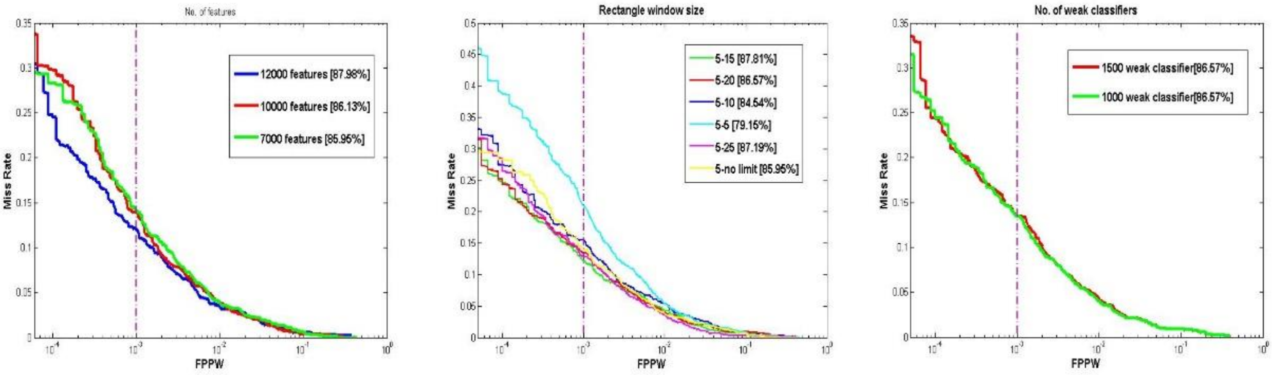


Fig. 5 Gradient Histogram Orientation along six orientation bins



(a) No. of features **(b) Rectangle Window Size** **(c) No. of Weak Classifiers**

Fig. 6 Detector Performance Evaluation

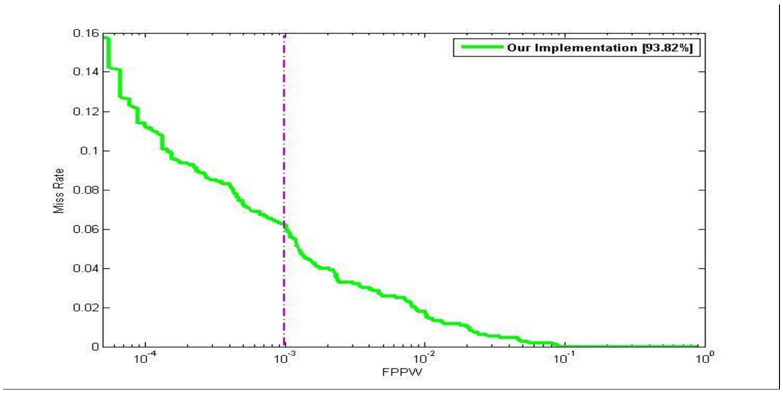


Fig. 7 Final Performance of the Detector

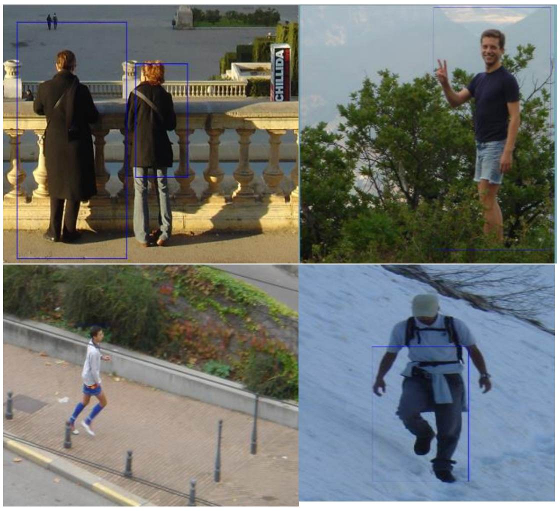


Fig. 8 Images with pedestrian

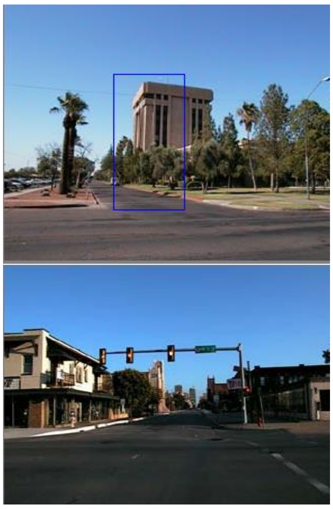


Fig. 9 Images without pedestrian