REAL-TIME PEDESTRIAN DETECTION USING EIGENFLOW

Dhiraj Goel, Tsuhan Chen

Carnegie Mellon University
Department of Electrical and Computer Engineering, Pittsburgh, USA
dhiraj@cmu.edu, tsuhan@cmu.edu

ABSTRACT

We propose a novel learning algorithm to detect moving pedestrians from a stationary camera in real-time. The algorithm learns a discriminative model based on eigenflow, i.e. the eigenvectors derived from applying Principal Component Analysis to the optical flow of moving objects, to differentiate between human motion patterns from other kind of motions like cars etc. The learned model is a cascade of Adaboost classifiers of increasing complexity, with eigenflow vectors as the weak classifiers. Unlike some recent attempts to use motion for pedestrian detection, this system performs this task in real-time. The system is also robust to small camera jitter and illumination changes. Moreover, we are able to detect moving children using the same system even though the training data is mainly composed of adult pedestrians.

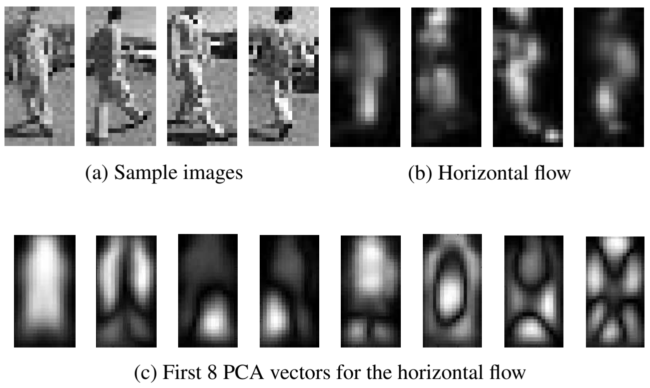
***Index Terms*—** Optical Flow, PCA, AdaBoost

1. INTRODUCTION AND RELATED WORK

Pedestrian detection is a hard problem in computer vision. Intra-class variability of the class of pedestrians makes the detection process very challenging. Pedestrians can appear in different poses and articulation due to the variations of the human body, especially the movement of the legs and the arms. Further, variations in clothing, background clutter, different lighting conditions and fluctuations in weather conditions render the problem extremely difficult.

Most of the pedestrian detection systems in the literature use appearance cues to detect pedestrians in a single image. Cues like wavelet response [1], histogram of oriented gradients [2] etc. have been used to learn a shape-based model to segment out human-like objects from a scene. However, appearance alone is not sufficient to detect humans in an uncontrolled outdoor environment.

Recently, there has been a lot of interest in using motion patterns to detect humans. Viola et al [3] used spatio-temporal filters based on shifted frame difference to augment the detection using spatial filters. Since dense optical flow is a popular method to represent motion, Fablet and Black [4] used it to learn a generative human-motion model while Hedvig [5] trained a Support Vector Machine to detect human-motion patterns. However, most of the optical flow-based detection methods are not real-time due to high computational cost. In this paper, we present a novel detection system that works in real-time.


(a) Sample images (b) Horizontal flow

(c) First 8 PCA vectors for the horizontal flow

Fig. 1. (a) Sample images from the pedestrian training data set (b) Corresponding horizontal flow of images in (a). (c) First eight PCA vectors for the horizontal flow component of the pedestrian motion patterns in the training data.

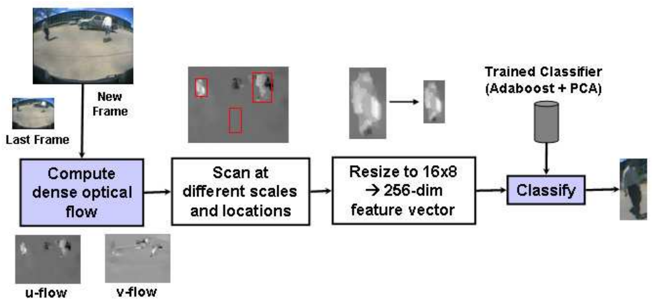
The rest of the paper is organized as follows: Section 2 describes the proposed method with emphasis on optical flow method employed (Section 2.1), the training stage (Section 2.2) and the detection method (Section 2.3), Section 3 evaluates the performance of the algorithm in different conditions and Section 4 concludes with a discussion and future work.

2. APPROACH

The pedestrian detection system that we propose learns to differentiate between human-like and non-human-like motion patterns. Fig 2 gives an overview of the system. The two main components of the system are computing real-time dense optical flow and learning the discriminative classifier.

2.1. Real-time Dense Optical Flow

Dense optical flow technique is a popular method to estimate motion between consecutive frames. Pioneering work in this



New Frame
Last Frame
Compute dense optical flow
Scan at different scales and locations
Resize to 16x8 → 256-dim feature vector
Trained Classifier (Adaboost + PCA)
Classify
u-flow
v-flow

Fig. 2. Overview of the proposed pedestrian detection

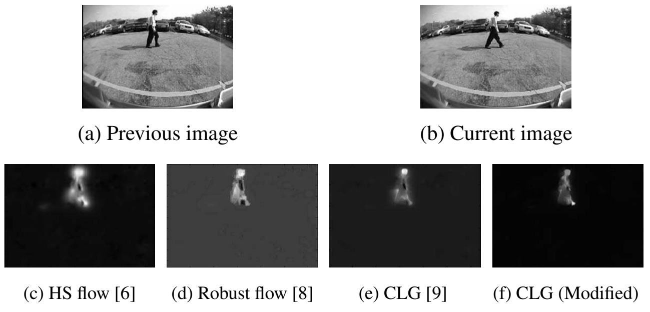


Fig. 3. Horizontal flow comparison for different optical flow algorithms.

field was done by Horn and Schunck [6]. Thereafter, several improvements have been proposed to get more accurate flow by either changing the weighting function of the regularization term in the global minimization condition [7] or posing the problem as a non-linear robust minimization instead of standard quadratic optimization [8]. However, most of these modifications are computationally expensive and hence, are not suitable for real-time applications.

Recently, Bruhn et al [9] proposed the use of multigrid methods to compute dense flow using Combined Local Global method (CLG). The weighting function used by them for real-time performance is:

$$w(x,y) = \frac{1}{\sqrt{1 + (f_x^2 + f_y^2) * \alpha}} \tag{1}$$

where f_x and f_y are the image gradients and α is a constant. Though such an image-driven weighting function is fast to compute, but is less accurate as it doesn't take into account the temporal information. To circumvent this problem, we propose a new spatio-temporal weighting function:

$$w(x,y) = \frac{1}{\sqrt{1 + (f_x^2 + f_y^2) * (\alpha * f_t^2 + \beta)}} \tag{2}$$

where f_t denotes the temporal gradient, and α and β are constants. Fig 3 shows a comparison of the performance of our modified CLG method with other methods.

2.2. Learning the Discriminative Classifier

Training Data: The training data for the pedestrian flow patterns was generated by hand-labelling moving pedestrians in the video sequences and then computing their optical flow. For the non-pedestrian motion patterns, the data-set was generated by scanning a variable sized window in the videos containing moving objects like cars, partial limbs etc but no complete moving human.

The training data consisted of 2400 pedestrian and non-pedestrian optical flow images each, with both horizontal and vertical components (Figure 1(a)(b)). Each of these was resized to 16x8 resolution using bilinear interpolation and normalized individually by dividing by the maximum magnitude u and v-flow. Finally, both the flow components were concatenated to give a 256-dimension motion pattern vector $x = [u_1, \ldots, u_{128}, v_1, \ldots, v_{128}]^T$ for each training data sample.

Weak Classifier: Principal Component Analysis was done separately on pedestrian and non-pedestrian data to obtain eigenflow [10]. Figure 1(c) shows the first few u-flow PCA vectors for the human motion. As in evident, the second PCA vector captures the motion of the hands/arms while the third and the fourth ones correspond to the right and left leg motion respectively. Using all the PCA vectors, 256 for each of the class, we have 512 eigenflow vectors that act as a pool of features for AdaBoost. Taking the magnitude of projection (invariant to the direction of motion) of a training data sample x onto a PCA vector z_j and finding the optimum threshold θ_j that minimizes the overall classification error would yield a weak classifier h_j.

$$h_j(x) = \begin{cases} 1, & \text{if } |x^T z_j| \leqslant \theta_j \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

Feature Selection and AdaBoost: The procedure to choose the most discriminative of these weak classifiers is motivated by the face detection algorithm proposed by Viola-Jones [11] and is described in Table 1. The final strong classifier is a weighted vote of the weak classifiers (Eqn. (4)). Figure 4(a) depicts the first two features selected by this algorithm. While the first one responds to motion near the boundary, the second one gives high value to motion within the window. Individually, they are weak but as a combination, they can perform much better (low value for the first feature and high for the second) and this is the basic idea that AdaBoost builds upon to learn a strong classifier. Figure 4 (b) shows the ROC curve for the strong classifier as number of weak classifiers are increased.

The advantage of such a classifier over more sophisticated classifiers like SVM is that it can provide comparable performance in significantly less time. Each of the weak classifier is just a dot product followed by thresholding which can be

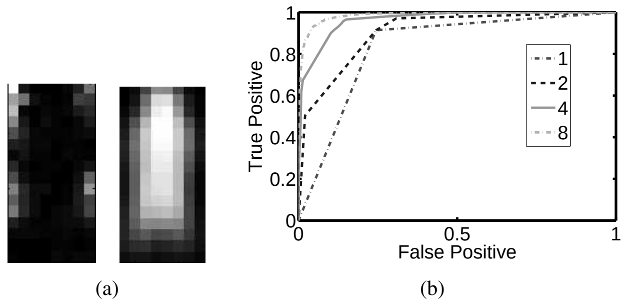


Fig. 4. (a) First two most discriminative PCA vectors. (b) ROC curve for Adaboost classifier as number of weak classifiers are increased.

Table 1. Feature selection and training AdaBoost classifier

- Given the training data $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$ where x_i is the eigenflow and y_i is 0 for non-pedestrian and 1 for pedestrian examples.
- Initialize the weights $w_{1,i} = \frac{1}{2l}, \frac{1}{2m}$ for $y_i = 0, 1$ respectively, where l and m are the number of pedestrian and non-pedestrian examples.
- for $t = 1, \ldots, T$
  1. Normalize the weights $w_{t,i} \leftarrow \frac{w_{t,i}}{\sum_{j=1}^{n} w_{t,j}}$
  2. Select the best weak classifier h_t with respect to the weighted error: $\epsilon_t = min_j \sum_i w_i |h_j - y_i|$
  3. Update the weights: $w_{t+1,i} = w_{t,i}\beta_t^{1-e_i}$ where $e_i = 0$ if example x_i is correctly classified by h_t, $e_i = 1$ otherwise, and $\beta_t = \frac{\epsilon_t}{1-\epsilon_t}$.
- The strong classifier is given by:

$$C(x) = \begin{cases} 1, & \sum_{t=1}^{T} \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^{T} \alpha_t \\ 0, & \text{otherwise.} \end{cases} \tag{4}$$

where $\alpha_t = \log \frac{1}{\beta_t}$

highly optimized using a platform like OpenCV.

Cascade of AdaBoost Classifiers: Even with AdaBoost, increasing the number of weak classifiers would hurt the real-time operation of the system while too few classifiers would compromise on the performance. An efficient way to get around this problem is to train a cascade of strong AdaBoost classifiers [11]. The ones at the beginning have fewer number of weak classifiers and hence, are really fast at classification. These are able to reject flow patterns that are highly unlikely to belong to humans but retain the ones that have some resemblance. Hence, the data points that pass these earlier stages need more complex analysis and this is where later stages of the cascade prove useful. To be labelled as a detection, a data sample has to pass all the stages in a cascade.

In our implementation, we have 7 stages in the cascade. The first classifier in the cascade has 4 weak classifiers and is able to reject approx. 40% of the non-pedestrian motion patterns while retaining almost all the pedestrian data. The second stage has 10 classifiers, third 15 and so on. The last stage has 100 classifiers. The pedestrian training data was the same for all the stages. The non-pedestrian data, for the next stage, was generated by running the current cascade of classifier on different training videos and collecting the false positives (maximum 2400).

2.3. Detecting Human Motion Patterns

Human motion patterns are detected in a video sequence using the procedure outlined in the Figure 2. The optical flow images are scanned using sub-windows of 7 different scales - 32x16, 96x48, 128x64, 160x80, 192x96, 256x128 and 320x160. The scanning window scales are always kept even multiples of 16x8 to allow fast downsampling. Each scale has an associated step size that increases with scale.

Camera position and orientation: The smaller sub windows are used to scan for far-off pedestrians while the larger sub-windows search for nearer ones. Without any knowledge about the camera geometry, the whole image has to be scanned for every scaled sub-window. However, knowing a priori, the position and the orientation of the camera would limit the scan range, esp. for the smaller scaled sub-windows. In the experimental results shown in Figure 5(a) and (b), exploiting such an information reduced the number of scanned windows by more than half.

Minimum flow criterion: In order to be reported as a detection, the queried sub-window must have non-zero optical flow. Thus, before resizing and feeding to the classifier, every sub-window must satisfy a minimum flow criterion. Since the pedestrians moving at a distance would appear to be moving slower as compared to closer pedestrians even if their actual speeds would say otherwise, the minimum optical flow thresholds vary according to scale. Furthermore, instead of having a single threshold for the whole sub-window, there are three thresholds, one each for three equal horizontal sub-regions within the sub-window. This rejects the sub-windows that have non-zero flow but the flow distribution doesn't conform to the erect human motion. All these thresholds were found from the pedestrian training data.

Once a sub-window satisfies the minimum flow criterion, it is resized to 16x8, normalized and fed to the cascade of classifiers. Multiple over-lapping detections are reported as one by taking their average.

3. EXPERIMENTS

The system was implemented in C++ using OpenCV libraries. On a 1.86 GHz Pentium M machine, we were able to achieve

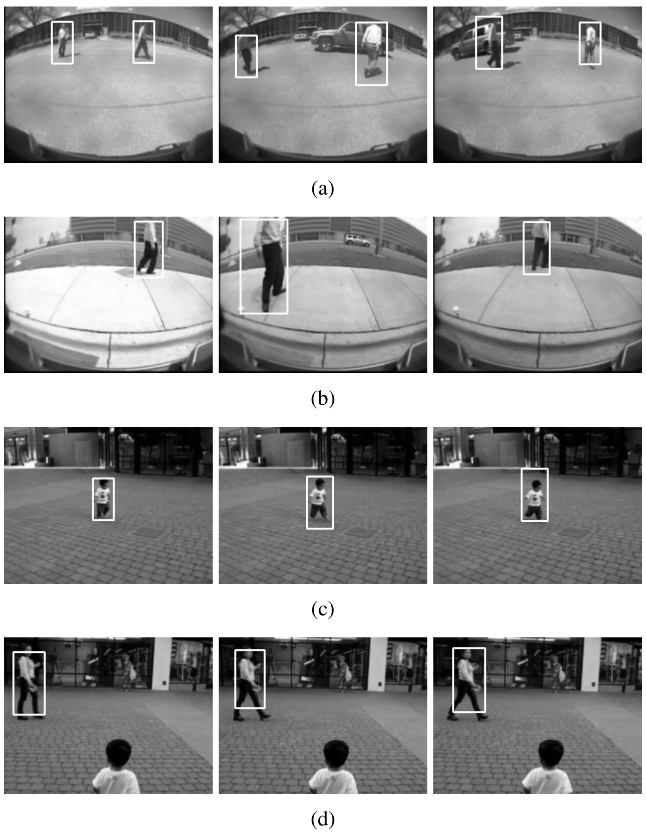


Fig. 5. Experimental results. (a) Detecting multiple humans in the presence of moving cars. (b) Detection while there's change in illumination. (c) Moving child detection. (d) Detection in the presence of small camera jitter.

real-time detection of moving pedestrians for 320x240 resolution grayscale video sequences at 10 fps.

Figure 5 shows some of the detection results in the test videos. The algorithm was tested with multiple moving humans in the presence of other moving objects, mainly cars and is able to detect humans in different poses and moving at different pace (Figure 5(a)). The false alarm rate was about one in every three frames but only if there are moving objects other than humans present in the scene. The miss rate is negligible for nearby pedestrians but the far-off pedestrians get rejected, at times, since their optical flow is too small. The system is also robust to illumination changes (Figure 5(b)), small camera jitter (Figure 5(d)) and can also detect moving children (Figure 5(c)). Such a system can be a great application for child safety, e.g. preventing accidents from reversing vehicles.

4. CONCLUSION AND FUTURE WORK

In this paper, we have proposed a novel pedestrian detection system by learning to discriminate between human-like motion patterns from other kind of motions like that of moving cars. The system has been shown to work in real-time and is robust to natural illumination changes and small camera jitter. Moreover, the applicability of such a framework towards detecting moving children has also been explored.

As future work, we plan to analyze the relationship between the global motion of a moving blob and its local intra-blob motion to reduce the false positives. Further, we plan to extend the current system to a moving camera by either warping the flow or by using its gradient.

Acknowledgement: *Work is supported in part by General Motors.*

5. REFERENCES

[1] M. Oren, C. Papageorgiou, P. Sinha, E. Osuna and T. Poggio, "Pedestrian detection using wavelet templates," in *CVPR*, 1997, pp. 193–199.

[2] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *CVPR*, 2005, vol. 1, pp. 886–893.

[3] P. Viola, M. Jones and D. Snow, "Detecting pedestrians using patterns of motion and appearance," *ICCV*, vol. 02, pp. 734–741, 2003.

[4] R. Fablet and M.J. Black, "Automatic detection and tracking of human motion with a view-based representation," in *ECCV*, 2002, vol. 1, pp. 476–491.

[5] H. Sidenbladh, "Detecting human motion with support vector machines," in *ICPR*, 2004, vol. 2, pp. 188–191.

[6] B.K.P. Horn, and B.G. Schunck, "Determining optical flow," *Artificial Intelligence*, vol. 17, pp. 185–203, 1981.

[7] H. Nagel and W. Enkelmann, "An investigation of smoothness constraints for the estimation of displacement vector fields from image sequences," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 8, no. 5, pp. 565–593, 1986.

[8] M. Black and P. Anandan, "The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields," *CVIU*, vol. 63, pp. 75–104, 1996.

[9] A. Bruhn, J. Weickert, T. Kohlberger, C. Schnörr, "A multigrid platform for real-time motion computation with discontinuity-preserving variational methods," *IJCV*, vol. 69, no. 2, pp. 257–277, 2006.

[10] X. Liu, T. Chen and B. Vijaya Kumar, "Face authentication for multiple subjects using eigenflow," *Pattern Recognition*, vol. 36, no. 2, pp. 313–328, 2003.

[11] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *CVPR*, 2001.