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 of 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 works in real-time. Moreover, the system is robust to small camera motion and slow illumination changes, and can detect moving children even though the training data had only adult pedestrians.

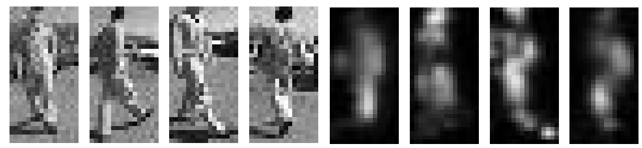
Index Terms— Optical Flow, PCA, AdaBoost

1. INTRODUCTION AND RELATED WORK

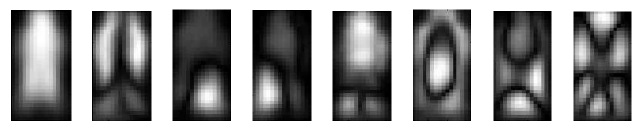
Pedestrian detection is a hard problem in computer vision. High intra-class variability of the pedestrians due to variations in pose, articulation and clothing makes the detection process very challenging. The background clutter, different lighting conditions and fluctuations in weather conditions add to further complications.

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 Support Vector Machines 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 (320x240 @ 10fps on Pentium M 1.86 GHz) .



(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: Sect. 2 describes the proposed method with emphasis on the optical flow method employed (Sect. 2.1), the training stage (Sect. 2.2) and the detection method (Sect. 2.3), Sect. 3 evaluates the performance of the algorithm in different conditions and Sect. 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. Figure 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 field was done by Horn and Schunck [6]. Thereafter, several improvements have been proposed to get more accurate flow

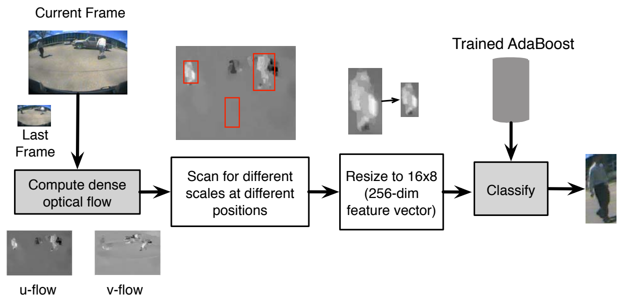
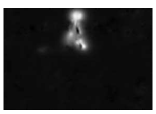


Fig. 2. Overview of the proposed pedestrian detection

(a) Previous image

(b) Current image



(c) HS flow [6]

(d) Robust flow [7]

(e) CLG [8]

(f) CLG (Modified)

Fig. 3. Horizontal flow computed using different optical flow algorithms.

by either changing the weighting function of the regularization term in the global minimization condition [9] or posing the problem as a non-linear robust minimization instead of standard quadratic optimization [7]. However, most of these modifications are computationally expensive and hence, are not suitable for real-time applications.

Recently, Bruhn et al [8] proposed the use of multigrid methods to compute dense flow using Combined Local Global method (CLG). The image-driven weighting function used by them was:

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

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

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

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

2.2. Learning the Discriminative Classifier

Training Data: The training data consisted of 2400 pedestrian and non-pedestrian optical flow patterns each, with both horizontal (Fig. 1(a),(b)) and vertical components. While the pedestrian data was generated by hand-labeling moving people in the images, the non-pedestrian data was collected by scanning a variable sized window in the videos containing moving objects like cars, partial limbs etc but no complete moving human.

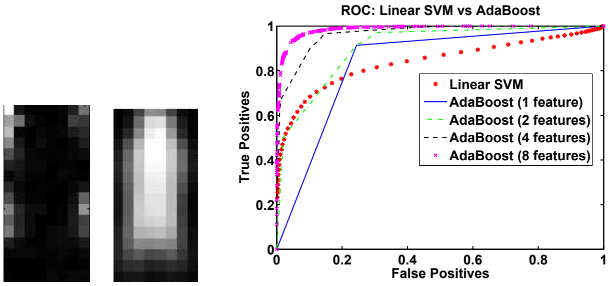
Each of these was resized to 16x8 resolution using bilinear interpolation and normalized individually to lie between $[-1, 1]$. Finally, both the flow components were concatenated to give a 256-dimension motion pattern vector $x = [u_1, \ldots, u_{128}, v_1, \ldots, v_{128}]^T$ for each training data 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 eigenvectors for the human motion. As is evident, the second eigenflow vector captures the motion of the hands/arms while the next two represent the leg motion. Collecting all eigenflow vectors, 256 for each class, we get a total of 512 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 an eigenflow vector z_j and finding the optimum threshold θ_j that minimizes the overall classification error yields a weak classifier h_j.

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

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 (Eq. (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 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 comparison with linear SMV classifier is also shown. Clearly, SVM is outperformed by choosing merely 4 weak classifiers for the Adaboost.

Cascade of AdaBoost Classifiers: Even with AdaBoost, increasing the number of weak classifiers would hurt the real-time operation of the system while selecting too few would compromise the detection accuracy. An efficient way to retain the advantages of being both fast and accurate is to train a cascade of AdaBoost classifiers [11]. Under this scheme,



(a) (b)

Fig. 4. (a) First two most discriminative PCA vectors. (b) Comparison of ROC curves: Linear SVM and Adaboost (with different number of weak classifiers or features).

the early stage classifiers 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. The flow patterns that pass these earlier stages need more complex analysis and this is where later stages of the cascade prove useful. To be labeled 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 collecting the false positives from the current classifier cascade (maximum 2400) from other videos.

2.3. Detecting Human Motion Patterns

Human motion patterns are detected in a video sequence using the procedure depicted in Fig. 2. The optical flow images are scanned using sub-windows of 7 different scales - 32x16, 48x24, 64x32, 80x40, 96x48, 128x64 and 160x80 - that are kept multiples of 16x8 to allow fast downsampling. Each scale has an associated step size that increases with scale to prevent excessive overlap between neighboring sub-windows.

Camera position and orientation: The smaller sub-windows scan for far-off pedestrians while the larger sub-windows search for ones closer to the camera. Without any knowledge about the camera geometry, an exhaustive search has be done in the whole image. However, knowing a priori, the position and the orientation of the camera can limit the scan range, since the pedestrians can be found only on the ground plane. In the experimental results shown in Fig. 5(a) and (b), exploiting such an information reduced the number of scanned windows by more than half.

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

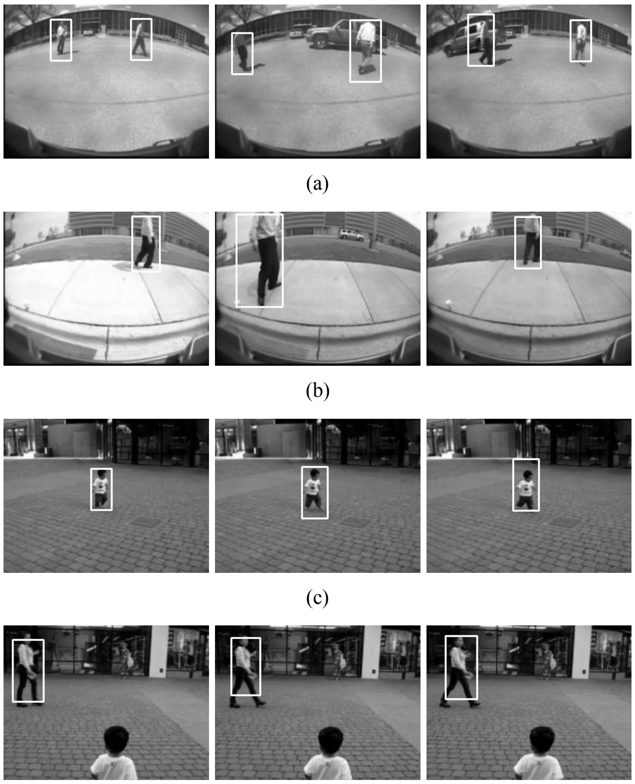
Minimum flow criterion: Every candidate sub-window must satisfy a minimum flow criterion before being tested against the classifier. Since, due to parallax, the far-off pedestrians would appear to be moving slower as compared to the nearer ones, the minimum optical flow thresholds vary with scale of the sub-window. Furthermore, instead of having a single threshold for the whole sub-window, three thresholds are employed, one each for three equal horizontal sub-regions within the sub-window. This rejects the regions 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.

If 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 merged to report a single detection.

3. EXPERIMENTS

The system was implemented in C++ using OpenCV libraries. On a 1.86 GHz Pentium M machine, we were able to detect moving pedestrians in 320x240 resolution grayscale image sequences at 10 fps. Computationally, a total of 429 dot products need to be computed for every candidate sub-window in the worst case. This is in sharp contrast to 13,000 dot products required per sub-window by linear SVM, one per support vector, resulting in an fps of 0.125.

Figure 5 shows some of the detection results in the test videos. The algorithm is able to detect humans in different



(a)

(b)

(c)

(d)

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 and small camera motion. (d) Detection in the presence of small camera motion.

poses and moving at different pace and reject other moving objects like cars (Fig. 5(a)). The false alarm rate was about one in every three frames when objects other than humans are present in the scene. The high false alarm rate is due to the large number of candidate sub-windows (about 1000) in each frame. 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 robust to slow illumination changes due to occasional cloud cover (Fig. 5(b)), small camera motion (Fig. 5(c),(d)) and can also detect moving children (Fig. 5(c)). Such a system can be a great application for child safety, e.g. preventing accidents from reversing vehicles. However, people moving in groups can be missed since the flow in candidate sub-window doesn't correspond to clean human motion.

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 with performance superior to that of a linear SVM and is robust to illumination changes and small camera motion. Moreover, the applicability of such a framework towards detecting moving children was also 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] M. Black and P. Anandan, "The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields," *CVIU*, vol. 63, pp. 75–104, 1996.

[8] 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.

[9] 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, pp. 565–593, 1986.

[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, pp. 511–518.