Human Detection Using Oriented Histograms of Flow and Appearance

Navneet Dalal, Bill Triggs, and Cordelia Schmid

GRAVIR-INRIA, 655 avenue de l'Europe, Montbonnot 38330, France
Firstname.Lastname@inrialpes.fr
http://lear.inrialpes.fr

Abstract. Detecting humans in films and videos is a challenging problem owing to the motion of the subjects, the camera and the background and to variations in pose, appearance, clothing, illumination and background clutter. We develop a detector for standing and moving people in videos with possibly moving cameras and backgrounds, testing several different motion coding schemes and showing empirically that orientated histograms of differential optical flow give the best overall performance. These motion-based descriptors are combined with our Histogram of Oriented Gradient appearance descriptors. The resulting detector is tested on several databases including a challenging test set taken from feature films and containing wide ranges of pose, motion and background variations, including moving cameras and backgrounds. We validate our results on two challenging test sets containing more than 4400 human examples. The combined detector reduces the false alarm rate by a factor of 10 relative to the best appearance-based detector, for example giving false alarm rates of 1 per 20,000 windows tested at 8% miss rate on our Test Set 1.

1 Introduction

Detecting humans in video streams is a challenging problem owing to variations in pose, body shape, appearance, clothing, illumination and background clutter. Moving cameras or backgrounds make it even harder. Potential applications include film and television analysis, on-line pedestrian detection for smart vehicles [8] and video surveillance. Although single-image appearance based detectors have made considerable advances in recent years (*e.g.* [3, 13, 15]), they are not yet reliable enough for many practical applications. On the other hand, certain kinds of movement are very characteristic of humans, so detector performance can potentially be improved by including motion information. Most existing work in this area assumes that the camera and the background are essentially static. This greatly simplifies the problem because the mere presence of motion already provides a strong cue for human presence. For example, Viola *et al.* [23] find that including motion features markedly increases the overall performance of their system, but they assume a fixed surveillance camera viewing a largely static scene. In our case, we wanted a detector that could be used to analyse film and TV content, or to detect pedestrians from a moving car – applications in which the camera and the background often move as much as the people in the scene, if not more. The main challenge is thus to find a set of features that characterize human motion well, while remaining resistant to camera and background motion.

A. Leonardis, H. Bischof, and A. Prinz (Eds.): ECCV 2006, Part II, LNCS 3952, pp. 428–441, 2006.

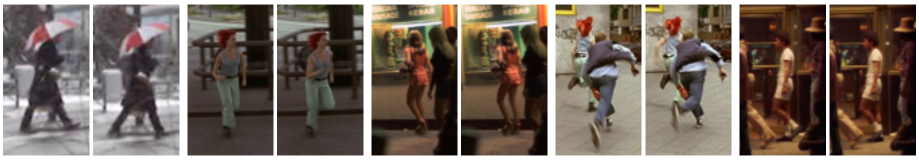


Fig. 1. Sample images from our human motion database, which contains moving people with significant variation in appearance, pose, clothing, background, illumination, coupled with moving cameras and backgrounds. Each pair shows two consecutive frames.

This paper introduces and evaluates a number of motion-based feature sets for human detection in videos. In particular it studies oriented histograms of various kinds of local differences or differentials of optical flow as motion features, evaluating these both independently and in combination with the Histogram of Oriented Gradient (HOG) appearance descriptors that we originally developed for human detection in static images [3]. The new descriptors are designed to capture the relative motion of different limbs while resisting background motions. Combining them with the appearance descriptors reduces the false alarm rate by an order of magnitude in images with movement while maintaining the performance of the original method [3] in stationary images.

The detectors are evaluated on two new and challenging feature film based data sets, giving excellent results. Fig. 1 shows some typical image pairs from our '*Test Set 1*' (see § 7).

Contents. § 2 briefly reviews the state-of-art in human detection in static and moving images. § 3 describes the overall system architecture. § 4–7 respectively describe the appearance descriptors, the motion descriptors, the optical flow methods and the training and test data sets that we used. § 8 studies the effect of representation choices and parameter settings on performance, and § 9 summarizes the results.

2 Previous Work

We will only mention a few of the more recent works on human detection here – see Gavrilla's survey [7] for older references. A polynomial SVM based pedestrian detector (upright whole-body human detector) using rectified Haar wavelets as input descriptors is described in [17] , with a parts (subwindow) based variant in [16]. The pedestrian detector of Gavrila & Philomen [9] takes a more direct approach, extracting edge images and matching them to a set of learned exemplars using chamfer distance. This has recently been extended to a practical real-time pedestrian detection system [8]. The success of SIFT appearance descriptors [14] for object recognition has motivated several recent approaches. Mikolajczyk *et al.* [15] use position-orientation histograms of binary image edges as image features, combining seven "part" (subwindow) detectors to build a static-image detector that is robust to occlusions. Our own static detector [3] uses a dense grid of SIFT-like blocks with a linear SVM for static-image person detection, giving false alarm rates 1–2 orders of magnitude lower than [17]. Leibe *et al.* [13] developed an effective static-image pedestrian detector for crowded scenes by coding local image patches against a learned codebook and combining the resulting bottom up labels with top-down refinement.

Regarding person detectors that incorporate motion descriptors, Viola *et al.* [23] build a detector for static-camera surveillance applications, using generalized Haar wavelets and block averages of spatiotemporal differences as image and motion features and a computationally efficient rejection chain classifier [1,22,21] trained with AdaBoost [19] feature selection. The inclusion of motion features increases the performance by an order of magnitude relative to a similar static detector. Other surveillance based detectors include the flow-based activity recognition system of Haritaoglu *et al.* [10]. Efros *et al.* [4] used appearance and flow features in an exemplar based detector for long shots of sports players, but quantitative performance results were not given.

3 Overall Architecture

This paper focuses on developing effective motion features so we have adopted a single relatively simple learning framework as a baseline in most of our experiments. For simplicity, we concentrate on detecting people who are upright and fully or almost fully visible. However they may be stationary or moving, against a background that may be stationary or moving. Linear SVM's [20] are used as a baseline classifier. They offer good performance relative to other linear classifiers and they are fast to run, providing at least a prospect of reliable real time detection. Three properties make them valuable for comparative testing work: reliable, repeatable training; the ability to handle large data sets gracefully; and good robustness to different choices of feature sets and parameters. Nonlinear SVM's typically achieve slightly lower error rates, but this comes at the cost of greatly increased run time and in practice we find that the main conclusions about feature sets remain unchanged.

Our person detector combines appearance descriptors extracted from a single frame of a video sequence with motion descriptors extracted from either optical flow or spatio-temporal derivatives against the subsequent frame. It scans a 64×128 pixel window across the image at multiple scales, running a linear SVM classifier on the descriptors extracted from each resulting image window. The classifier is trained to make person/no-person decisions using a set of manually labeled training windows. Fig. 2 gives an overview of the feature extraction process. Image gradient vectors are used to produce weighted votes for local gradient orientation and these are locally histogrammed to produce an appearance descriptor (SIFT / HOG process) [3]. Differentials of optical flow are fed to a similar oriented voting process based on either flow orientation or oriented

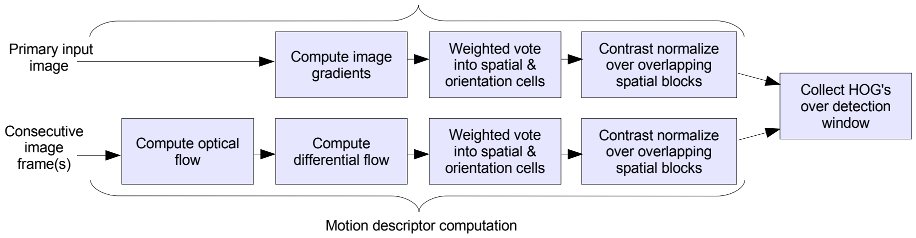

Appearance descriptor computation
Primary input image
Compute image gradients
Weighted vote into spatial & orientation cells
Contrast normalize over overlapping spatial blocks
Collect HOG's over detection window
Consecutive image frame(s)
Compute optical flow
Compute differential flow
Weighted vote into spatial & orientation cells
Contrast normalize over overlapping spatial blocks
Motion descriptor computation

Fig. 2. The feature extraction process for our combined detector

spatial gradients of flow components. Each descriptor set is normalized over local, overlapping blocks of spatial cells, and the resulting normalized histograms are concatenated to make the detection window descriptor vector used in the detector.

For the learning process we use a method similar to that of [3]. We start with a set of training images (here consecutive image pairs so that flow can be used) in which all of the positive training windows (ones containing people) have been manually marked. A fixed set of initial negative training windows was selected by randomly sampling the negative images. A preliminary classifier is trained on the marked positives and initial negatives, and this is used to search the complete set of negative images exhaustively for false alarms. As many of these “hard negatives” as will fit into the available RAM are selected randomly and added to the training set, and the final classifier is trained. Each classifier thus has its own set of hard negatives. This retraining procedure significantly increases the performance of every detector that we have tested. Additional rounds of search for hard negatives make little difference, so are not used. In most of the experiments below the RAM is limited to 1.5 GB, so the larger the descriptor vector, the smaller the number of hard examples that can be included. We think that this is fair as memory is typically the main resource limitation during training.

In use, the algorithm runs a detection window across the image at all positions and scales, giving a detection score at each point. Negative scores are zeroed and a 3D position-scale mean shift process [2] is run to identify significant local peaks in the resulting score. If above threshold, these are declared as detections. Currently there is no attempt to enforce temporal continuity of detections: the detector runs independently in each pair of images.

4 Appearance Descriptors

The static-image part of our descriptor set [3] uses Histogram of Oriented Gradient grids (HOG) – a close relation of the descriptor in Lowe’s SIFT approach [14] – to code visual appearance. Briefly, the HOG method tiles the detector window with a dense grid of cells, with each cell containing a local histogram over orientation bins. At each pixel, the image gradient vector is calculated and converted to an angle, voting into the corresponding orientation bin with a vote weighted by the gradient magnitude. Votes are accumulated over the pixels of each cell. The cells are grouped into blocks and a robust normalization process is run on each block to provide strong illumination invariance. The normalized histograms of all of the blocks are concatenated to give the window-level visual descriptor vector for learning. To reduce aliasing, spatial and angular linear interpolation, and in some cases Gaussian windowing over the block, are used during voting. The blocks overlap spatially so that each cell appears several times with different normalizations, as this typically improves performance. See [3] for further details and a study of the effects of the various parameters. The same default parameter settings are used here.

5 Motion Descriptors

To use motion for human detection from moving cameras against dynamic backgrounds we need features that characterize human movements well while remaining resistant to typical camera and background motions. Most of the existing motion descriptors, such as the phase based features of Fleet & Jepson [5] and the generalized wavelet features of Viola *et al.* [23], use absolute motions and hence work well only when the camera and background are largely static. Nor do these representations take into account the lessons learned from the SIFT / HOG family of descriptors [14,15,3]. This section introduces descriptors that use differential flow to cancel out most of the effects of camera motion and HOG like oriented histogram voting to obtain robust coding.

First note that the image flow induced by camera rotation (pan, tilt, roll) varies smoothly across the image irrespective of 3D depth boundaries, and in most applications it is locally essentially translational because significant camera roll is rare. Thus, any kind of local differential or difference of flow cancels out most of the effects of camera rotation. The remaining signal is due to either depth-induced motion parallax between the camera, subject and background, or to independent motion in the scene. Differentials of parallax flows are concentrated essentially at 3D depth boundaries, while those of independent motions are largest at motion boundaries. For human subjects, both types of boundaries coincide with limb and body edges, so flow differentials are good cues for the outline of a person. However we also expect internal dynamics such as relative limb motions to be quite discriminant for human motions and differentials taken within the subject's silhouette are needed to capture these. Thus, flow-based features can focus either on coding motion (and hence depth) boundaries, or on coding internal dynamics and relative displacements of the limbs.

Notation. $\mathcal{I}^x, \mathcal{I}^y$ denote images containing the x (horizontal) and y (vertical) components of optical flow, $\mathcal{I}^{\mathbf{w}} = (\mathcal{I}^x, \mathcal{I}^y)$ denote the 2D flow image ($\mathbf{w} = (x, y)$), and $\mathcal{I}^x_x, \mathcal{I}^x_y, \mathcal{I}^y_x, \mathcal{I}^y_y$ denote the corresponding x- and y-derivative differential flow images. *E.g.*, $\mathcal{I}^x_y = \frac{d}{dy}\mathcal{I}^x$ is the y-derivative of the x component of optical flow.

5.1 Motion Boundary Based Coding

For motion boundary coding it is natural to try to capture the local orientations of motion edges by emulating the static-image HOG descriptors [3]. The simplest approach is to treat the two flow components $\mathcal{I}^x, \mathcal{I}^y$ as independent 'images', take their local gradients separately, find the corresponding gradient magnitudes and orientations, and use these as weighted votes into local orientation histograms in the same way as for the standard gray scale HOG. We call this family of schemes *Motion Boundary Histograms (MBH)* (see Fig. 3). A separate histogram can be built for each flow component, or the two channels can be combined, *e.g.* by the winner-takes-all voting method used to handle color channels in [3]. We find that separate histograms are more discriminant. As with standard gray scale HOG, it is best to take spatial derivatives at the smallest possible scale ($[1, 0, -1]$ mask) without any form of smoothing.

5.2 Internal / Relative Dynamics Based Coding

One could argue that the static appearance descriptor already captures much of the available boundary information, so that the flow based descriptor should focus more on capturing complementary information about internal or relative motions. This suggests that flow differences should be computed between pairs of nearby, but not necessarily

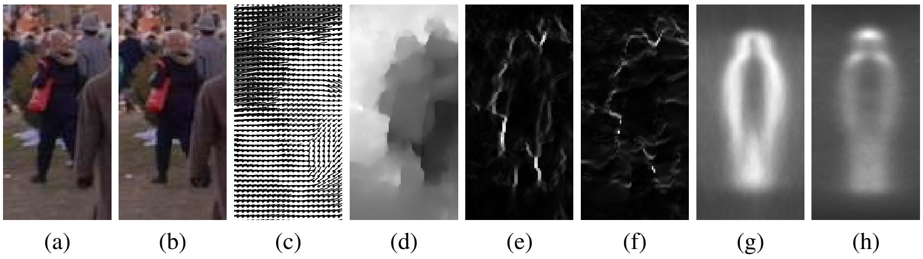


Fig. 3. Illustration of the MBH descriptor. (a,b) Reference images at time t and $t+1$. (c,d) Computed optical flow, and flow magnitude showing motion boundaries. (e,f) Gradient magnitude of flow field $\mathcal{I}^x$, $\mathcal{I}^y$ for image pair (a,b). (g,h) Average MBH descriptor over all training images for flow field $\mathcal{I}^x$, $\mathcal{I}^y$.

neighboring, points, and that angular voting should be based on the direction of the flow difference vector, not the direction of the spatial derivative displacement. So in opposition to MBH, we use $(\mathcal{I}^x_x, \mathcal{I}^y_x)$ and $(\mathcal{I}^x_y, \mathcal{I}^y_y)$ as the pairs for angular voting, and the simple x, y derivatives are replaced by spatial differences taken at larger scales, perhaps in several different directions. We will call this family of schemes Internal Motion Histograms (IMH). Ideally, IMH descriptors would directly capture the relative movements of different limbs, *e.g.* left *vs.* right leg, but choosing the necessary spatial displacements for differencing would require reliable part detectors. Instead we test simple variants based on fixed spatial displacements, as follows:

IMHdiff is the simplest IMH descriptor. It takes fine-scale derivatives, using $(\mathcal{I}^x_x, \mathcal{I}^y_x)$ and $(\mathcal{I}^x_y, \mathcal{I}^y_y)$ to create two relative-flow-direction based oriented histograms. As with MBH, using separate orientation histograms for the x- and y-derivatives is better than combining them. Variants of IMHdiff use larger (but still central) spatial displacements for differencing – 5 pixels apart ($[1, 0, 0, 0, -1]$ mask), or even 7 – and take spatial differencing steps along several different directions, *e.g.* including diagonal axes.

IMHcd uses the blocks-of-cells structure of the HOG descriptors differently. It uses 3×3 blocks of cells, in each of the 8 outer cells computing flow differences for each pixel relative to the corresponding pixel in the central cell and histogramming to give an orientation histogram[1]. Figure 4(a) illustrates. The resulting 8 histograms are normalized as a block. The motivation is that if the person's limb width is approximately the same as the cell size, IMHcd can capture relative displacements of the limbs w.r.t. to the background and nearby limbs. The results in § 8 support this hypothesis.

IMHmd is similar to IMHcd, but instead of using the corresponding pixel in the central cell as a reference flow, it uses the average of the corresponding pixels in all 9 cells. The resulting 9 histograms are normalized as a block.

IMHwd is also similar to IMHcd but uses Haar wavelet like operators rather than non-central differences, as shown in Fig. 4(b).

[1] IMHcd uses non-central cell-width spatial differences that access only pixels within the block, whereas IMHdiff uses central differences and in the boundary cells it accesses pixels that lie outside the block.

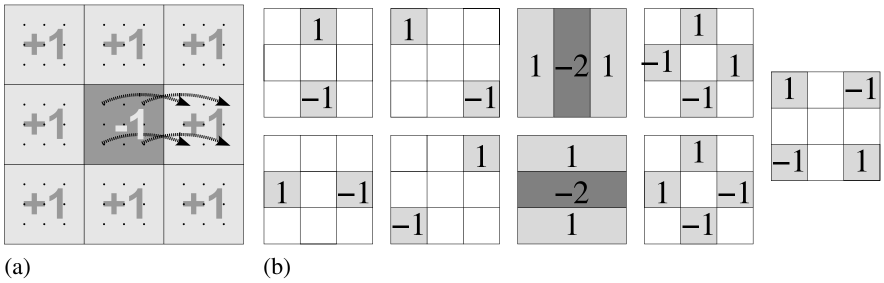


Fig. 4. Different coding schemes for IMH descriptors. (a) One block of IMHcd coding scheme. The block is partitioned into cells. The dots in each cell represent the cell pixels. The arrows emerging from the central cell show the central pixel used to compute differences for the corresponding pixel in the neighbouring cell. Similar differences are computed for each of the 8 neighbouring cells. Values $+1$ and -1 represent the difference weights. (b) The wavelet operators used in the IMHwd motion coding scheme.

ST Diff. We also evaluated a scheme inspired by Viola *et al.* [23] based on simple spatiotemporal differencing rather than flow. For each pixel, its $3{\times}3$ stride-8 neighborhood at the next time step is taken and its image intensity is subtracted from each of these 9 pixels. The absolute values are accumulated over each cell to make a 9 bin histogram for the cell, which then undergoes the usual block normalization process.

5.3 Descriptor Parameters

For the combined flow and appearance detectors with the optimal cell size of $8{\times}8$ pixels, memory constraints limit us to a total of about 81 histogram bins per cell. (Increasing the histogram size beyond this is possible, but it reduces the number of hard negatives that can be fitted into memory during re-training to such an extent that performance suffers). In the experiments below, we test: MBH with 9 gradient orientations, 2 separate flow components, and $4\times$ block overlap; IMHdiff with 2 displacements (horizontal and vertical $[1, 0, -1]$ masks), 9 flow orientations and $4\times$ block overlap; and IMHcd, IMHwd and IMHmd with eight 8-pixel displacements and 6 flow orientations.

All of the methods use orientation histograms with votes weighted by vector modulus followed by a block-level normalization – essentially the same scheme as the original HOG descriptor [3]. We tested various different bin sizes, normalization schemes, *etc.* with similar conclusions to [3]. For both MBH and IMHdiff, fine (9 bin) orientation coding with $2{\times}2$ blocks of $8{\times}8$ pixel cells seem to be best. 3×3 blocks of cells ($9\times$ block overlap) perform better for the flow-only MBH classifier, but for the combined detectors the performance of this combination drops owing to the increased feature size. Changing the cell size from 8×8 to 6×6 only reduces the performance slightly. Good normalization of the blocks is critical and for the flow descriptors Lowe's hysteresis-based L2 normalization seems to do significantly better than L2 or L1-sqrt normalization. We tried larger displacement masks (3- and 5- pixel displacement) for MBH but found that the performance drops. For the IMHcd/wd/md schemes, 6 and 9 orientation bins give the same performance (we use 6 below), and Lowe's hysteresis based L2 normalization still works best, but only by a small margin.

We also evaluated variants that use the least squares image prediction error of the estimated flow as a flow quality metric, down-weighting the histogram vote in proportion to $\exp(-|e|/\sigma)$, where e is the fitting error over the local 5×5 window. This very slightly ($\lesssim 1\%$) improves the performance provided that σ is not set too small.

We also tested various motion descriptors that do not use orientation voting (*e.g.* based simply on the modulus of velocity), but the results were significantly worse.

6 Optical Flow Estimation

We tried several optical flow methods. Our initial testing was done with the Otago implementation [6] of the Proesmans *et al.* [18] multi-scale nonlinear diffusion based algorithm. This gives dense high-quality sub-pixel motion estimates but it is computationally expensive (15 seconds per frame). Also, motion boundaries are critical for human detection and we recently began to suspect that the Otago flows were over-regularized for this application. To test this we implemented a simple but fast flow method based on the constant brightness assumption [11]. Flow is found top-down in a multi-scale approach, with initial flow estimates made at a coarse scale propagated downwards and refined in fine scale steps. The flow $\mathbf{w}$ is estimated *independently* at each pixel by solving a damped Linear Least Squares equation $\mathbf{w} = (\mathbf{A}^\top\mathbf{A} + \beta\mathbf{I})^{-1}\mathbf{A}^\top\mathbf{b}$ over a small $N \times N$ neighborhood, where $\mathbf{b}$ is an N^2 column vector encoding the temporal image differences, $\mathbf{A}$ is an $N^2 \times 2$ matrix of spatial gradients $[\mathcal{I}_x, \mathcal{I}_y]$, and β is a damping factor included to reduce numerical issues arising from singular $\mathbf{A}^\top\mathbf{A}$. The model does not include any explicit spatial regularization or smoothing and its flow estimates are visibly less accurate than the Otago ones, but our experiments show that using it in the combined detector reduces false positives by a factor of more than 3 at 8% miss rate. In fact, any regularization aimed at improving the flow smoothness appears to reduce the detector performance. Our method is also much faster than the Otago one, running in 1 second on DVD resolution 752×396 images, with $N = 5$ and a scale refinement step of 1.3. The new method is used in all of the experiments in § 8 unless otherwise noted.

We also tested motion descriptors based on an MPEG-4 block matcher taken from the `www.xvid.org` codec. No attempt was made to enforce motion continuity between blocks. Even though the matching estimates were visually good, the detection results were not competitive. We think that there are several reasons for this. Firstly, block matching provides only one vote for each cell, whereas with optical flow each pixel provides a separate vote into the histogram. Secondly, the block matching flow estimates do not have deep sub-pixel accuracy. Experiments on rounding the flow values from the Otago code showed that even 1/10 of a pixel of rounding causes the performance to drop significantly (the need for accurate orientation voting is one reason for this). Thirdly, 8×8 MPEG blocks are too large for the best results.

7 Data Sets

To train our detectors, we selected shots from various movie DVDs and personal digital camera video sequences and annotated the humans in them. Our main training set, '*Training Set 1*', was obtained from 5 different DVDs. It contains a total of 182 shots with 2781 human examples (5562 including left-right reflections). We created two test sets. '*Test Set 1*' contains 50 shots and 1704 human examples from unseen shots from the DVDs used in *Training Set 1*. *Test Set 2* is more challenging, containing 2700 human examples from 128 shots from 6 new DVDs.

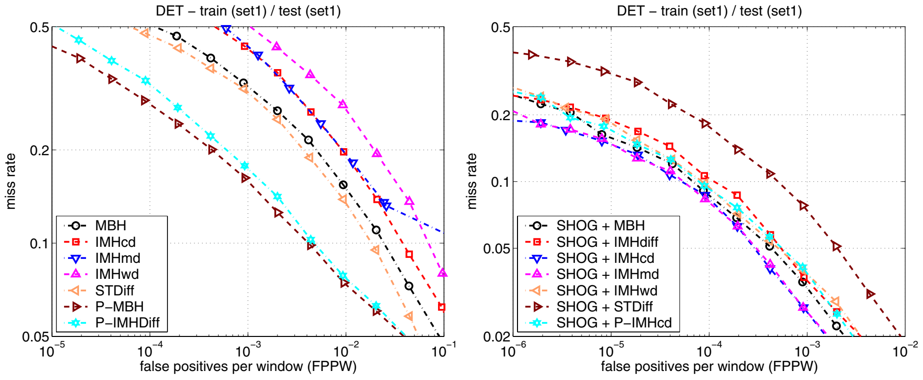
We have also used the static-image training and test sets from [3] (available at `http://pascal.inrialpes.fr/data/human/`). In this paper, we call these the '*Static Training/Test Sets*'. They contain respectively 2416 training and 1132 test images. Even though the *Static Training Set* has no (zero) flow, we find that including it along with *Training Set 1* significantly improves the performance of both the static and the combined detectors (see § 8). More precisely, the detector performance on *Test Set 1* improves, without changing that of the static detector on the *Static Test Set*. This is perhaps because the *Set 1* images contain many poses that do not appear in the *Static* sets – notably running and other rapid actions.

8 Experiments

To quantify the performance of the various detectors we plot Detection Error Tradeoff (DET) curves, *i.e.* Miss Rate ($1 - \text{Precision}$ or $N_{\text{FalseNeg}}/(N_{\text{FalseNeg}} + N_{\text{TruePos}})$) versus False Positives Per Window tested (FPPW) on logarithmic scales. DET plots present the same information as Receiver Operating Characteristic (ROC) curves in a more readable form. Lower curves are better.

We begin by comparing the results of the motion descriptors introduced above, trained and tested on Set 1. Figure 5(a,b) give results respectively for detectors learned with the motion descriptors alone, and for detectors that include both these features and the HOG appearance descriptors. The oriented histogram of differential flow schemes MBH and IMHdiff with the Proesmans flow method dominate the motion-only results. In fact for the video test sets (which do contain many frames without much visible movement) these motion features alone are within an order of magnitude of the static HOG detector and significantly better than the static Haar wavelet detector. When motion and appearance features are combined, neither the Proesmans flow method nor the MBH descriptors perform so well and it is IMHcd and IMHmd computed using our flow method that are the leaders. Below we use SHOG + IMHcd as the default combined detector, although SHOG + IMHmd would lead to similar conclusions.

Fig. 5 shows that motion-only results are not a good guide to the performance of the combined detector. The reduced spread of the results in the combined case suggests that there is a considerable degree of redundancy between the appearance and motion channels. In particular, IMHdiff and MBH are the schemes with the smallest spatial strides and thus the greatest potential for redundancy with the human boundary cues used by the appearance based descriptors – factors that may explain their reduced performance after combination. Similarly, the strong regularization of the Proesmans' flow estimates may make them effective cues for motion (and hence occlusion) boundaries, while the unregularized nature of ours means that they capture motion within thin limbs more accurately and hence provide information that is more complementary to the appearance descriptors.



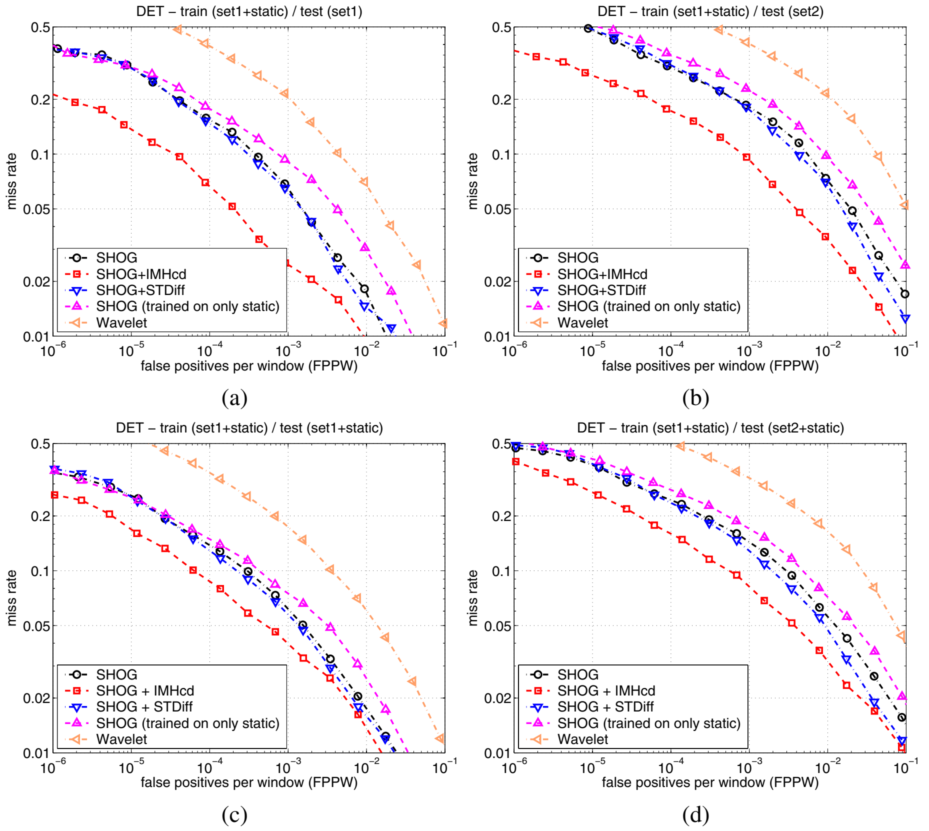
(a) (b)

Fig. 5. A comparison of the different motion descriptors, trained on *Training Set 1* and tested on *Test Set 1*, using: (a) the motion feature set alone; and (b) the motion feature set combined with the SHOG appearance descriptor. The prefix 'P' in the MBH and IMH legends denotes the same methods using Proesmans' flow estimates.

Figure 6 demonstrates the overall performance of a selection of our detectors on several different test sets. Unless otherwise noted, the detectors are trained on the combined *Set 1* and *Static* Training Sets. The static (appearance based) detectors shown are: SHOG – the HOG detector of [3]; SHOG (static) – SHOG trained on the *Static Training Set* alone, as in [3]; and Wavelet – our version of the static Haar wavelet based detector of [17]. Two combined detectors are also shown: SHOG + IMHcd – SHOG combined with the IMHcd flow feature (8-pixel steps in 8-neighbor directions); and SHOG + ST Diff – SHOG combined with Viola *et al.* spatiotemporal differences [23].

Again the good performance of the SHOG + IMHcd combination is apparent. The absolute results on Test Set 2 are an order of magnitude worse than on Test Set 1 owing to the more challenging nature of the images, but the relative rankings of the different methods are remarkably stable. Overall, on video data for which motion estimates are available, the false alarm rates of the best combined detectors are an order of magnitude lower than those for the best static-appearance-based ones.

Given that we want methods that can detect people reliably whether or not they are moving, we were concerned that the choice of method might be sensitive to the relative proportion of moving and of static people in the videos. To check this, we tested the detectors not only on the pure video *Test Sets 1* and *2*, but also on the combination of these with the *Static Test Set* (again with static image flows being zero). The results are shown in fig. 6(c–d). Diluting the fraction of moving examples naturally reduces the advantage of the combined methods relative to the static ones, but the relative ranking of the methods remains unchanged. Somewhat surprisingly, table 1 shows that when used on entirely static images for which there is no flow, the best combined detectors do marginally *better* the best static one. The images here are from the *Static Test Set*, with the detectors trained on *Training Set 1* plus the *Static Training Set* as before.



(a)

(b)

(c)

(d)

Fig. 6. An overview of the performance of our various detectors. All detectors are trained on *Training Set 1* combined with the *Static Training Set* with flow set to zero. They are tested respectively on: (a) *Test Set 1*; (b) *Test Set 2*; (c) *Test Set 1* plus the *Static Test Set*; (d) *Test Set 2* plus the *Static Test Set*.

Table 1. The miss rates of various detectors trained on Set 1 + Static images and tested on purely Static images. Despite the complete lack of flow information, the combined detectors provide slightly better performance than the static one.

FPPW	10^{-3}	10^{-4}	10^{-5}
SHOG	6.2%	11.4%	19.8%
SHOG + IMHcd	5.8%	11.0%	19.8%
SHOG + ST Diff	5.7%	10.5%	19.7%

Figure 7 shows some sample detections of the combined detector (SHOG + IMHcd trained on Set 1 + Static) on images from *Test Set 2*. *Set 2* contains challenging images taken from different films from the training images. Here there are shots of people in Indian costume, some dance sequences, and people in crowds that are different from anything seen in the training images.

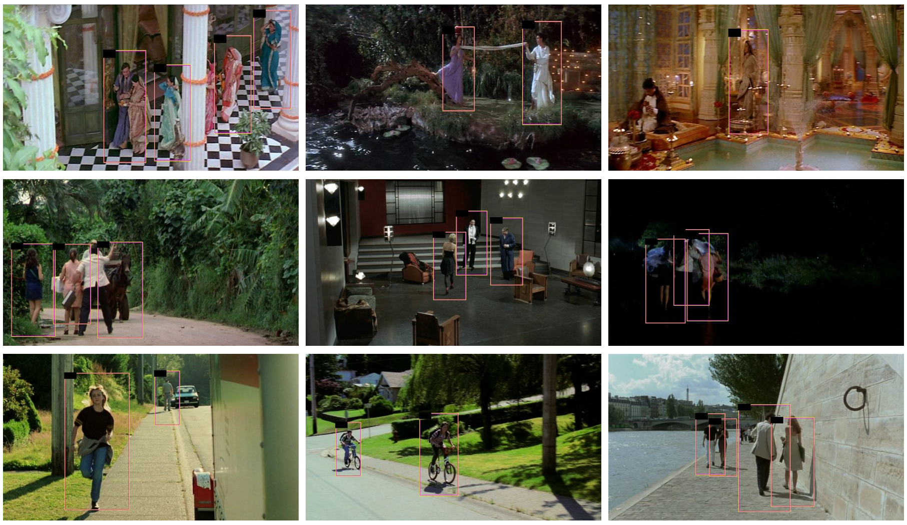


Fig. 7. Sample detections on *Test Set 2* from the combined SHOG + IMHcd detector trained on *Set 1* + *Static*. Note the variations in pose, appearance, background and lightning.

The experiments shown here use linear SVMs. Informal tests with Gaussian kernel SVMs suggest that these would reduce false positives by an additional factor of about 2, at a cost of a 5000-fold increase in run time.

Mixture of Experts. The combined-feature detectors above are *monolithic* – they concatenate the motion and appearance features into a single large feature vector and train a combined classifier on it. We have also tested an alternative *Mixture of Experts* architecture. In this, separate detectors are learned from the appearance features and from the motion features, and a second stage classifier is then trained to combine the (real valued scalar) outputs of these to produce a combined detector. In our case the second stage classifier is a linear SVM over a 2D feature space (the appearance score and the motion score), so the final system remains linear in the input features. This approach keeps the feature space dimensions relatively low during training, thus allowing more hard negatives to be included at each stage. (Indeed, for the 2D second stage classifier there can be millions of them). In our experiments these effects mitigate the losses due to separate training and the linear Mixture of Experts classifier actually performs slightly better than the best monolithic detector. For now the differences are marginal (less than 1%), but the Mixture of Experts architecture provides more flexibility and may ultimately be preferable. The component classifiers could also be combined in a more sophisticated way, for example using a rejection cascade [1,22,21] to improve the runtime.

9 Summary and Conclusions

We have developed a family of high-performance detectors for fully visible humans in videos with moving people, cameras and backgrounds. The detectors combine gradient based appearance descriptors with differential optical flow based motion descriptors in a linear SVM framework. Both motion and appearance channels use oriented histogram voting to achieve a robust descriptor. We studied various different motion coding schemes but found that although there are considerable performance differences between them when motion features alone are used, the differences are greatly reduced when the features are used in combination with static appearance descriptors. The best combined schemes used motion descriptors based on oriented histogramming of differences of unregularized multiscale flow relative to corresponding pixels in adjacent cells (IMHcd) or to local averages of these (IMHmd).

Acknowledgments. This work was supported by the European Union research projects ACEMEDIA and PASCAL. SVMLight [12] proved reliable for training large-scale SVM's. We thank Matthijs Douze for his comments on the manuscript.

References

[1] S. Baker and S. Nayar. Pattern rejection. In *Proceedings of the Conference on Computer Vision and Pattern Recognition, San Francisco, California, USA*, 1996.

[2] D. Comaniciu. An algorithm for data-driven bandwidth selection. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 25(2):281–288, 2003.

[3] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proceedings of the Conference on Computer Vision and Pattern Recognition, San Diego, California, USA*, pages 886–893, 2005.

[4] A. Efros, A. Berg, G. Mori, and J. Malik. Recognizing action at a distance. In *Proceedings of the 9th International Conference on Computer Vision, Nice, France*, pages II:726–733, 2003.

[5] D. Fleet and A. Jepson. Stability of phase information. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 15(12):1253–1268, 1993.

[6] B. Galvin, B. McCane, K. Novins, D. Mason, and S. Mills. Recovering motion fields: An evaluation of eight optical flow algorithms. In *Proceedings of the ninth British Machine Vision Conference, Southampton, England*, 1998. http://www.cs.otago.ac.nz/research/vision.

[7] D. M. Gavrila. The visual analysis of human movement: A survey. *Computer Vision and Image Understanding*, 73(1):82–98, 1999.

[8] D. M. Gavrila, J. Giebel, and S. Munder. Vision-based pedestrian detection: the protector+ system. In *Proc. of the IEEE Intelligent Vehicles Symposium, Parma, Italy*, 2004.

[9] D. M. Gavrila and V. Philomin. Real-time object detection for smart vehicles. In *Proceedings of the Conference on Computer Vision and Pattern Recognition, Fort Collins, Colorado, USA*, pages 87–93, 1999.

[10] I. Haritaoglu, D. Harwood, and L. Davis. W4: Real-time surveillance of people and their activities. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 22(8):809–830, 2000.

[11] K.P. Horn and G. Schunck. Determining optical flow. *Artificial Intelligence*, 17:185–203, 1981.

[12] T. Joachims. Making large-scale svm learning practical. In B. Schlkopf, C. Burges, and A. Smola, editors, *Advances in Kernel Methods - Support Vector Learning*. The MIT Press, Cambridge, MA, USA, 1999.

[13] B. Leibe, E. Seemann, and B. Schiele. Pedestrian detection in crowded scenes. In *Proceedings of the Conference on Computer Vision and Pattern Recognition, San Diego, California, USA*, pages 876–885, June 2005.

[14] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[15] K. Mikolajczyk, C. Schmid, and A. Zisserman. Human detection based on a probabilistic assembly of robust part detectors. In *Proceedings of the 8th European Conference on Computer Vision, Prague, Czech Republic*, volume I, pages 69–81, 2004.

[16] A. Mohan, C. Papageorgiou, and T. Poggio. Example-based object detection in images by components. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 23(4): 349–361, April 2001.

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

[18] M. Proesmans, L. Van Gool, E. Pauwels, and A. Oosterlinck. Determination of optical flow and its discontinuities using non-linear diffusion. In *Proceedings of the 3rd European Conference on Computer Vision, Stockholm, Sweden*, volume 2, pages 295–304, 1994.

[19] R. E. Schapire. The boosting approach to machine learning, an overview. In *MSRI Workshop on Nonlinear Estimation and Classification*, 2002.

[20] Bernhard Schölkopf and Alex Smola. *Learning with Kernels*. The MIT Press, Cambridge, MA, USA, 2002.

[21] J. Sun, J.M. Rehg, and A. Bobick. Automatic cascade training with perturbation bias. In *Proceedings of the Conference on Computer Vision and Pattern Recognition, Washington, DC, USA*, pages II:276–283, 2004.

[22] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proceedings of the Conference on Computer Vision and Pattern Recognition, Kauai, Hawaii, USA*, volume I, pages 511–518, 2001.

[23] P. Viola, M. J. Jones, and D. Snow. Detecting pedestrians using patterns of motion and appearance. In *Proceedings of the 9th International Conference on Computer Vision, Nice, France*, volume 1, pages 734–741, 2003.