Histograms of Oriented Gradients for Human Detection

Navneet Dalal and Bill Triggs
INRIA Rhône-Alps, 655 avenue de l'Europe, Montbonnot 38334, France
{Navneet.Dalal,Bill.Triggs}@inrialpes.fr, http://lear.inrialpes.fr

Abstract

We study the question of feature sets for robust visual object recognition, adopting linear SVM based human detection as a test case. After reviewing existing edge and gradient based descriptors, we show experimentally that grids of Histograms of Oriented Gradient (HOG) descriptors significantly outperform existing feature sets for human detection. We study the influence of each stage of the computation on performance, concluding that fine-scale gradients, fine orientation binning, relatively coarse spatial binning, and high-quality local contrast normalization in overlapping descriptor blocks are all important for good results. The new approach gives near-perfect separation on the original MIT pedestrian database, so we introduce a more challenging dataset containing over 1800 annotated human images with a large range of pose variations and backgrounds.

1 Introduction

Detecting humans in images is a challenging task owing to their variable appearance and the wide range of poses that they can adopt. The first need is a robust feature set that allows the human form to be discriminated cleanly, even in cluttered backgrounds under difficult illumination. We study the issue of feature sets for human detection, showing that locally normalized Histogram of Oriented Gradient (HOG) descriptors provide excellent performance relative to other existing feature sets including wavelets [17,22]. The proposed descriptors are reminiscent of edge orientation histograms [4,5], SIFT descriptors [12] and shape contexts [1], but they are computed on a dense grid of uniformly spaced cells and they use overlapping local contrast normalizations for improved performance. We make a detailed study of the effects of various implementation choices on detector performance, taking "pedestrian detection" (the detection of mostly visible people in more or less upright poses) as a test case. For simplicity and speed, we use linear SVM as a baseline classifier throughout the study. The new detectors give essentially perfect results on the MIT pedestrian test set [18,17], so we have created a more challenging set containing over 1800 pedestrian images with a large range of poses and backgrounds. Ongoing work suggests that our feature set performs equally well for other shape-based object classes.

We briefly discuss previous work on human detection in §2, give an overview of our method §3, describe our data sets in §4 and give a detailed description and experimental evaluation of each stage of the process in §5–6. The main conclusions are summarized in §7.

2 Previous Work

There is an extensive literature on object detection, but here we mention just a few relevant papers on human detection [18,17,22,16,20]. See [6] for a survey. Papageorgiou *et al* [18] describe a pedestrian detector based on a polynomial SVM using rectified Haar wavelets as input descriptors, with a parts (subwindow) based variant in [17]. Depoortere *et al* give an optimized version of this [2]. Gavrila & Philomen [8] take a more direct approach, extracting edge images and matching them to a set of learned exemplars using chamfer distance. This has been used in a practical real-time pedestrian detection system [7]. Viola *et al* [22] build an efficient moving person detector, using AdaBoost to train a chain of progressively more complex region rejection rules based on Haar-like wavelets and space-time differences. Ronfard *et al* [19] build an articulated body detector by incorporating SVM based limb classifiers over 1^{st} and 2^{nd} order Gaussian filters in a dynamic programming framework similar to those of Felzenszwalb & Huttenlocher [3] and Ioffe & Forsyth [9]. Mikolajczyk *et al* [16] use combinations of orientation-position histograms with binary-thresholded gradient magnitudes to build a parts based method containing detectors for faces, heads, and front and side profiles of upper and lower body parts. In contrast, our detector uses a simpler architecture with a single detection window, but appears to give significantly higher performance on pedestrian images.

3 Overview of the Method

This section gives an overview of our feature extraction chain, which is summarized in fig. 1. Implementation details are postponed until §6. The method is based on evaluating well-normalized local histograms of image gradient orientations in a dense grid. Similar features have seen increasing use over the past decade [4,5,12,15]. The basic idea is that local object appearance and shape can often be characterized rather well by the distribution of local intensity gradients or

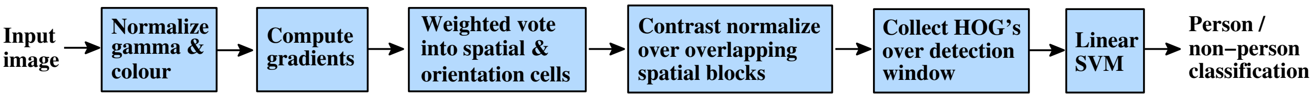


Figure 1. An overview of our feature extraction and object detection chain. The detector window is tiled with a grid of overlapping blocks in which Histogram of Oriented Gradient feature vectors are extracted. The combined vectors are fed to a linear SVM for object/non-object classification. The detection window is scanned across the image at all positions and scales, and conventional non-maximum suppression is run on the output pyramid to detect object instances, but this paper concentrates on the feature extraction process.

edge directions, even without precise knowledge of the corresponding gradient or edge positions. In practice this is implemented by dividing the image window into small spatial regions ("*cells*"), for each cell accumulating a local 1-D histogram of gradient directions or edge orientations over the pixels of the cell. The combined histogram entries form the representation. For better invariance to illumination, shadowing, *etc*., it is also useful to contrast-normalize the local responses before using them. This can be done by accumulating a measure of local histogram "energy" over somewhat larger spatial regions ("*blocks*") and using the results to normalize all of the cells in the block. We will refer to the normalized descriptor blocks as *Histogram of Oriented Gradient (HOG)* descriptors. Tiling the detection window with a dense (in fact, overlapping) grid of HOG descriptors and using the combined feature vector in a conventional SVM based window classifier gives our human detection chain (see fig. 1).

The use of orientation histograms has many precursors [13,4,5], but it only reached maturity when combined with local spatial histogramming and normalization in Lowe's *Scale Invariant Feature Transformation (SIFT)* approach to wide baseline image matching [12], in which it provides the underlying image patch descriptor for matching scale-invariant keypoints. SIFT-style approaches perform remarkably well in this application [12,14]. The *Shape Context* work [1] studied alternative cell and block shapes, albeit initially using only edge pixel counts without the orientation histogramming that makes the representation so effective. The success of these sparse feature based representations has somewhat overshadowed the power and simplicity of HOG's as dense image descriptors. We hope that our study will help to rectify this. In particular, our informal experiments suggest that even the best current keypoint based approaches are likely to have false positive rates at least 1–2 orders of magnitude higher than our dense grid approach for human detection, mainly because none of the keypoint detectors that we are aware of detect human body structures reliably.

The HOG/SIFT representation has several advantages. It captures edge or gradient structure that is very characteristic of local shape, and it does so in a local representation with an easily controllable degree of invariance to local geometric and photometric transformations: translations or rotations make little difference if they are much smaller that the local spatial or orientation bin size. For human detection, rather coarse spatial sampling, fine orientation sampling and strong local photometric normalization turns out to be the best strategy, presumably because it permits limbs and body segments to change appearance and move from side to side quite a lot provided that they maintain a roughly upright orientation.

4 Data Sets and Methodology

Datasets. We tested our detector on two different data sets. The first is the well-established MIT pedestrian database [18], containing 509 training and 200 test images of pedestrians in city scenes (plus left-right reflections of these). It contains only front or back views with a relatively limited range of poses. Our best detectors give essentially perfect results on this data set, so we produced a new and significantly more challenging data set, 'INRIA', containing 1805 64×128 images of humans cropped from a varied set of personal photos. Fig. 2 shows some samples. The people are usually standing, but appear in any orientation and against a wide variety of background image including crowds. Many are bystanders taken from the image backgrounds, so there is no particular bias on their pose. The database is available from *http://lear.inrialpes.fr/data* for research purposes.

Methodology. We selected 1239 of the images as positive training examples, together with their left-right reflections (2478 images in all). A fixed set of 12180 patches sampled randomly from 1218 person-free training photos provided the initial negative set. For each detector and parameter combination a preliminary detector is trained and the 1218 negative training photos are searched exhaustively for false positives ('hard examples'). The method is then re-trained using this augmented set (initial 12180 + hard examples) to produce the final detector. The set of hard examples is subsampled if necessary, so that the descriptors of the final training set fit into 1.7 Gb of RAM for SVM training. This retraining process significantly improves the performance of each detector (by 5% at 10^{-4} False Positives Per Window tested (FPPW) for our default detector), but additional rounds of retraining make little difference so we do not use them.

To quantify detector performance we plot *Detection Error Tradeoff (DET)* curves on a log-log scale, *i.e.* miss rate ($1-\mathrm{Recall}$ or $\frac{\mathrm{FalseNeg}}{\mathrm{TruePos}+\mathrm{FalseNeg}}$) versus FPPW. Lower values are better. DET plots are used extensively in speech and in NIST evaluations. They present the same information as Receiver Operating Characteristics (ROC's) but allow small

Figure 2. Some sample images from our new human detection database. The subjects are always upright, but with some partial occlusions and a wide range of variations in pose, appearance, clothing, illumination and background.

probabilities to be distinguished more easily. We will often use miss rate at 10^{-4}FPPW as a reference point for results. This is arbitrary but no more so than, *e.g*. Area Under ROC. In a multiscale detector it corresponds to a raw error rate of about 0.8 false positives per 640×480 image tested. (The full detector has an even lower false positive rate owing to non-maximum suppression). Our DET curves are usually quite shallow so even very small improvements in miss rate are equivalent to large gains in FPPW at constant miss rate. For example, for our default detector at 1e-4 FPPW, every 1% absolute (9% relative) reduction in miss rate is equivalent to reducing the FPPW at constant miss rate by a factor of 1.57.

5 Overview of Results

Before presenting our detailed implementation and performance analysis, we compare the overall performance of our final HOG detectors with that of some other existing methods. Detectors based on rectangular (R-HOG) or circular log-polar (C-HOG) blocks and linear or kernel SVM are compared with our implementations of the Haar wavelet, PCA-SIFT, and shape context approaches. Briefly, these approaches are as follows:

Generalized Haar Wavelets. This is an extended set of oriented Haar-like wavelets similar to (but better than) that used in [17]. The features are rectified responses from 9×9 and 12×12 oriented 1$^{\text{st}}$ and 2$^{\text{nd}}$ derivative box filters at $45°$ intervals and the corresponding 2$^{\text{nd}}$ derivative xy filter.

PCA-SIFT. These descriptors are based on projecting gradient images onto a basis learned from training images using PCA [11]. Ke & Sukthankar found that they outperformed SIFT for key point based matching, but this is controversial [14]. Our implementation uses 16×16 blocks with the same derivative scale, overlap, *etc*., as our HOG descriptors. The PCA basis is calculated using the positive training images.

Shape Contexts. The original Shape Contexts [1] used binary edge-presence voting into log-polar spaced bins, irrespective of edge orientation. We simulate this using our C-HOG descriptor (see below) with just 1 orientation bin. 16 angular and 3 radial intervals with inner radius 2 pixels and outer radius 8 pixels gave the best results. Both gradient-strength and edge-presence based voting were tested, with the edge threshold chosen automatically to maximize detection performance (the values selected were somewhat variable, in the region of 20–50 graylevels).

Results. Fig. 3 shows the performance of the various detectors on the MIT and INRIA data sets. The HOG-based detectors greatly outperform the wavelet, PCA-SIFT and Shape Context ones, giving near-perfect separation on the MIT test set and at least an order of magnitude reduction in FPPW on the INRIA one. Our Haar-like wavelets outperform MIT wavelets because we also use 2$^{\text{nd}}$ order derivatives and contrast normalize the output vector. Fig. 3(a) also shows MIT's best parts based and monolithic detectors (the points are interpolated from [17]), however beware that an exact comparison is not possible as we do not know how the database in [17] was divided into training and test parts and the negative images used are not available. The performances of the final rectangular (R-HOG) and circular (C-HOG) detectors are very similar, with C-HOG having the slight edge. Augmenting R-HOG with primitive bar detectors (oriented 2$^{\text{nd}}$ derivatives – 'R2-HOG') doubles the feature dimension but further improves the performance (by 2% at 10^{-4} FPPW). Replacing the linear SVM with a Gaussian kernel one improves performance by about 3% at 10^{-4} FPPW, at the cost of much higher run times[1]. Using binary edge voting (EC-HOG) instead of gradient magnitude weighted voting (C-HOG) decreases performance by 5% at 10^{-4} FPPW, while omitting orientation information decreases it by much more, even if additional spatial or radial bins are added (by 33% at 10^{-4} FPPW, for both edges (E-ShapeC) and gradients (G-ShapeC)). PCA-SIFT also performs poorly. One reason is that, in comparison to [11], many more (80 of 512) principal vectors have to be retained to capture the same proportion of the variance. This may be because the spatial registration is weaker when there is no keypoint detector.

6 Implementation and Performance Study

We now give details of our HOG implementations and systematically study the effects of the various choices on detector performance. Throughout this section we refer results

[1] We use the hard examples generated by *linear* R-HOG to train the kernel R-HOG detector, as kernel R-HOG generates so few false positives that its hard example set is too sparse to improve the generalization significantly.

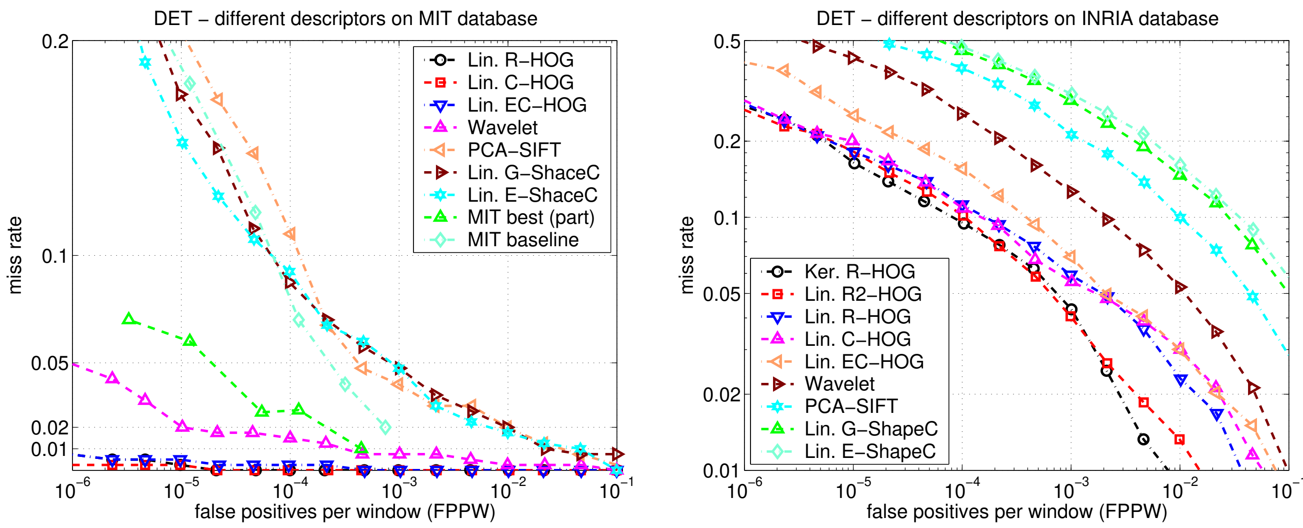


Figure 3. The performance of selected detectors on (left) MIT and (right) INRIA data sets. See the text for details.

to our default detector which has the following properties, described below: RGB colour space with no gamma correction; $[-1, 0, 1]$ gradient filter with no smoothing; linear gradient voting into 9 orientation bins in $0°$–$180°$; $16{\times}16$ pixel blocks of four $8{\times}8$ pixel cells; Gaussian spatial window with $\sigma = 8$ pixel; *L2-Hys* (Lowe-style clipped L2 norm) block normalization; block spacing stride of 8 pixels (hence 4-fold coverage of each cell); $64{\times}128$ detection window; linear SVM classifier.

Fig. 4 summarizes the effects of the various HOG parameters on overall detection performance. These will be examined in detail below. The main conclusions are that for good performance, one should use fine scale derivatives (essentially no smoothing), many orientation bins, and moderately sized, strongly normalized, overlapping descriptor blocks.

6.1 Gamma/Colour Normalization

We evaluated several input pixel representations including grayscale, RGB and LAB colour spaces optionally with power law (gamma) equalization. These normalizations have only a modest effect on performance, perhaps because the subsequent descriptor normalization achieves similar results. We do use colour information when available. RGB and LAB colour spaces give comparable results, but restricting to grayscale reduces performance by 1.5% at 10^{-4} FPPW. Square root gamma compression of each colour channel improves performance at low FPPW (by 1% at 10^{-4} FPPW) but log compression is too strong and worsens it by 2% at 10^{-4} FPPW.

6.2 Gradient Computation

Detector performance is sensitive to the way in which gradients are computed, but the simplest scheme turns out to be the best. We tested gradients computed using Gaussian smoothing followed by one of several discrete derivative masks. Several smoothing scales were tested including σ=0 (none). Masks tested included various 1-D point derivatives (uncentred $[-1, 1]$, centred $[-1, 0, 1]$ and cubic-corrected $[1, -8, 0, 8, -1]$) as well as $3{\times}3$ Sobel masks and $2{\times}2$ diagonal ones $\begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}$, $\begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}$ (the most compact centred 2-D derivative masks). Simple 1-D $[-1, 0, 1]$ masks at σ=0 work best. Using larger masks always seems to decrease performance, and smoothing damages it significantly: for Gaussian derivatives, moving from σ=0 to σ=2 reduces the recall rate from 89% to 80% at 10^{-4} FPPW. At σ=0, cubic corrected 1-D width 5 filters are about 1% worse than $[-1, 0, 1]$ at 10^{-4} FPPW, while the $2{\times}2$ diagonal masks are 1.5% worse. Using uncentred $[-1, 1]$ derivative masks also decreases performance (by 1.5% at 10^{-4} FPPW), presumably because orientation estimation suffers as a result of the x and y filters being based at different centres.

For colour images, we calculate separate gradients for each colour channel, and take the one with the largest norm as the pixel's gradient vector.

6.3 Spatial / Orientation Binning

The next step is the fundamental nonlinearity of the descriptor. Each pixel calculates a weighted vote for an edge orientation histogram channel based on the orientation of the gradient element centred on it, and the votes are accumulated into orientation bins over local spatial regions that we call *cells*. Cells can be either rectangular or radial (log-polar sectors). The orientation bins are evenly spaced over $0°$–$180°$ ("unsigned" gradient) or $0°$–$360°$ ("signed" gradient). To reduce aliasing, votes are interpolated bilinearly between the neighbouring bin centres in both orientation and position. The vote is a function of the gradient magnitude at the pixel, either the magnitude itself, its square, its square root, or a clipped form of the magnitude representing soft presence/absence of an edge at the pixel. In practice, using the magnitude itself gives the best results. Taking the square root

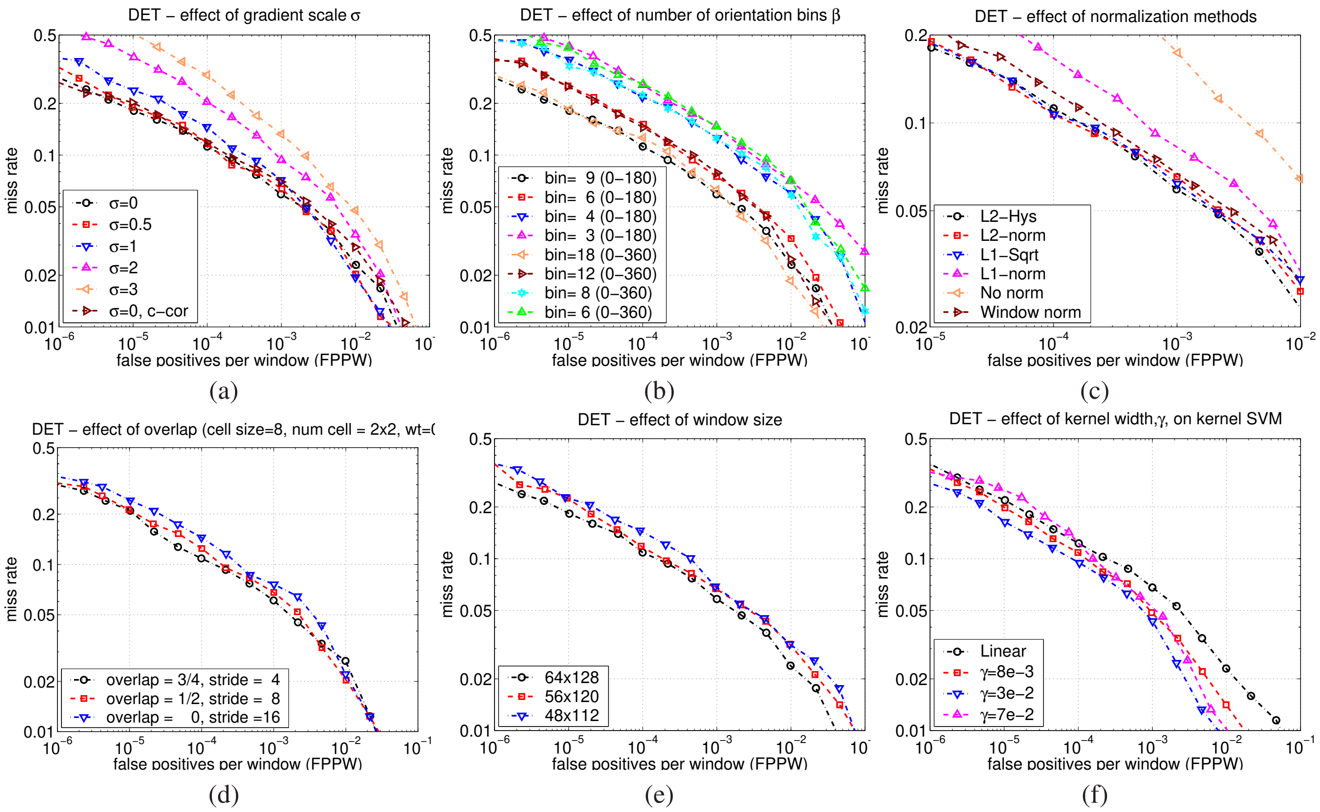


Figure 4. For details see the text. (a) Using fine derivative scale significantly increases the performance. ('c-cor' is the 1D cubic-corrected point derivative). (b) Increasing the number of orientation bins increases performance significantly up to about 9 bins spaced over $0°$–$180°$. (c) The effect of different block normalization schemes (see §6.4). (d) Using overlapping descriptor blocks decreases the miss rate by around 5%. (e) Reducing the 16 pixel margin around the 64×128 detection window decreases the performance by about 4%. (f) Using a Gaussian kernel SVM, $\exp(-\gamma\|\mathbf{x_1}-\mathbf{x_2}\|^2)$, improves the performance by about 3%.

reduces performance slightly, while using binary edge presence voting decreases it significantly (by 5% at 10^{-4} FPPW).

Fine orientation coding turns out to be essential for good performance, whereas (see below) spatial binning can be rather coarse. As fig. 4(b) shows, increasing the number of orientation bins improves performance significantly up to about 9 bins, but makes little difference beyond this. This is for bins spaced over $0°$–$180°$, *i.e.* the 'sign' of the gradient is ignored. Including signed gradients (orientation range $0°$–$360°$, as in the original SIFT descriptor) decreases the performance, even when the number of bins is also doubled to preserve the original orientation resolution. For humans, the wide range of clothing and background colours presumably makes the signs of contrasts uninformative. However note that including sign information does help substantially in some other object recognition tasks, *e.g.* cars, motorbikes.

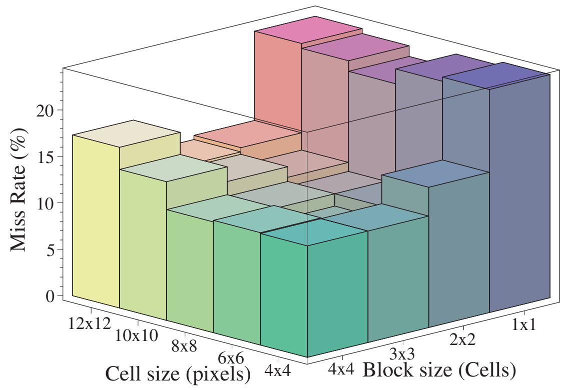


Figure 5. The miss rate at 10^{-4} FPPW as the cell and block sizes change. The stride (block overlap) is fixed at half of the block size. 3×3 blocks of 6×6 pixel cells perform best, with 10.4% miss rate.

6.4 Normalization and Descriptor Blocks

Gradient strengths vary over a wide range owing to local variations in illumination and foreground-background contrast, so effective local contrast normalization turns out to be essential for good performance. We evaluated a number of different normalization schemes. Most of them are based on grouping cells into larger spatial blocks and contrast normalizing each block separately. The final descriptor is then the vector of all components of the normalized cell responses from all of the blocks in the detection window. In fact, we typically overlap the blocks so that each scalar cell response contributes several components to the final descriptor vector, each normalized with respect to a different block. This may seem redundant but good normalization is critical and including overlap significantly improves the performance. Fig. 4(d) shows that performance increases by 4% at 10^{-4} FPPW as we increase the overlap from none (stride 16) to 16-fold area / 4-fold linear coverage (stride 4).

We evaluated two classes of block geometries, square or rectangular ones partitioned into grids of square or rectangular spatial cells, and circular blocks partitioned into cells in log-polar fashion. We will refer to these two arrangements as R-HOG and C-HOG (for rectangular and circular HOG).

R-HOG. R-HOG blocks have many similarities to SIFT descriptors [12] but they are used quite differently. They are computed in dense grids at a single scale without dominant orientation alignment and used as part of a larger code vector that implicitly encodes spatial position relative to the detection window, whereas SIFT's are computed at a sparse set of scale-invariant key points, rotated to align their dominant orientations, and used individually. SIFT's are optimized for sparse wide baseline matching, R-HOG's for dense robust coding of spatial form. Other precursors include the edge orientation histograms of Freeman & Roth [4]. We usually use square R-HOG's, *i.e.* $\varsigma\times\varsigma$ grids of $\eta\times\eta$ pixel cells each containing β orientation bins, where ς, η, β are parameters.

Fig. 5 plots the miss rate at 10^{-4} FPPW w.r.t. cell size and block size in cells. For human detection, 3×3 cell blocks of 6×6 pixel cells perform best, with 10.4% miss-rate at 10^{-4} FPPW. Our standard 2×2 cell blocks of 8×8 cells are a close second. In fact, 6–8 pixel wide cells do best irrespective of the block size – an interesting coincidence as human limbs are about 6–8 pixels across in our images. 2×2 and 3×3 cell blocks work best. Adaptivity to local imaging conditions is weakened when the block becomes too big, and when it is too small (1×1 cell block, *i.e.* normalization over orientation alone) valuable spatial information is suppressed.

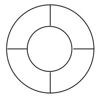
As in [12], it is useful to downweight pixels near the edges of the block by applying a Gaussian spatial window to each pixel before accumulating orientation votes into cells. This improves performance by 1% at 10^{-4} FPPW for a Gaussian with $\sigma = 0.5 * \mathrm{block\_width}$.

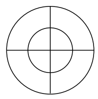
We also tried including multiple block types with different cell and block sizes in the overall descriptor. This slightly improves performance (by around 3% at 10^{-4} FPPW), at the cost of greatly increased descriptor size.

Besides square R-HOG blocks, we also tested vertical (2×1 cell) and horizontal (1×2 cell) blocks and a combined descriptor including both vertical and horizontal pairs. Vertical and vertical+horizontal pairs are significantly better than horizontal pairs alone, but not as good as 2×2 or 3×3 cell blocks (1% worse at 10^{-4} FPPW).

C-HOG. Our circular block (C-HOG) descriptors are reminiscent of Shape Contexts [1] except that, crucially, each spatial cell contains a stack of gradient-weighted orientation cells instead of a single orientation-independent edge-presence count. The log-polar grid was originally suggested by the idea that it would allow fine coding of nearby structure to be combined with coarser coding of wider context, and the fact that the transformation from the visual field to the V1 cortex in primates is logarithmic [21]. However small descriptors with very few radial bins turn out to give the best performance, so in practice there is little inhomogeneity or context. It is probably better to think of C-HOG's simply as an advanced form of centre-surround coding.

We evaluated two variants of the C-HOG geometry, ones with a single circular central cell (similar to the GLOH feature of [14]), and ones whose central cell is divided into angular sectors as in shape contexts. We present results only for the circular-centre variants, as these have fewer spatial cells than the divided centre ones and give the same performance in practice. A technical report will provide further details. The C-HOG layout has four parameters: the numbers of angular and radial bins; the radius of the central bin in pixels; and the expansion factor for subsequent radii. At least two radial bins (a centre and a surround) and four angular bins (quartering) are needed for good performance. Including additional radial bins does not change the performance much, while increasing the number of angular bins decreases performance (by 1.3% at 10^{-4} FPPW when going from 4 to 12 angular bins). 4 pixels is the best radius for the central bin, but 3 and 5 give similar results. Increasing the expansion factor from 2 to 3 leaves the performance essentially unchanged. With these parameters, neither Gaussian spatial weighting nor inverse weighting of cell votes by cell area changes the performance, but combining these two reduces slightly. These values assume fine orientation sampling. Shape contexts (1 orientation bin) require much finer spatial subdivision to work well.





Block Normalization schemes. We evaluated four different block normalization schemes for each of the above HOG geometries. Let $\mathbf{v}$ be the unnormalized descriptor vector, $\|\mathbf{v}\|_k$ be its k-norm for k=1, 2, and ϵ be a small constant. The schemes are: (a) *L2-norm*, $\mathbf{v} \rightarrow \mathbf{v}/\sqrt{\|\mathbf{v}\|_2^2 + \epsilon^2}$; (b) *L2-Hys*, L2-norm followed by clipping (limiting the maximum values of $\mathbf{v}$ to 0.2) and renormalizing, as in [12]; (c) *L1-norm*, $\mathbf{v} \rightarrow \mathbf{v}/(\|\mathbf{v}\|_1 + \epsilon)$; and (d) *L1-sqrt*, L1-norm followed by square root $\mathbf{v} \rightarrow \sqrt{\mathbf{v}/(\|\mathbf{v}\|_1 + \epsilon)}$, which amounts to treating the descriptor vectors as probability distributions and using the Bhattacharya distance between them. Fig. 4(c) shows that L2-Hys, L2-norm and L1-sqrt all perform equally well, while simple L1-norm reduces performance by 5%, and omitting normalization entirely reduces it by 27%, at 10^{-4} FPPW. Some regularization ϵ is needed as we evaluate descriptors densely, including on empty patches, but the

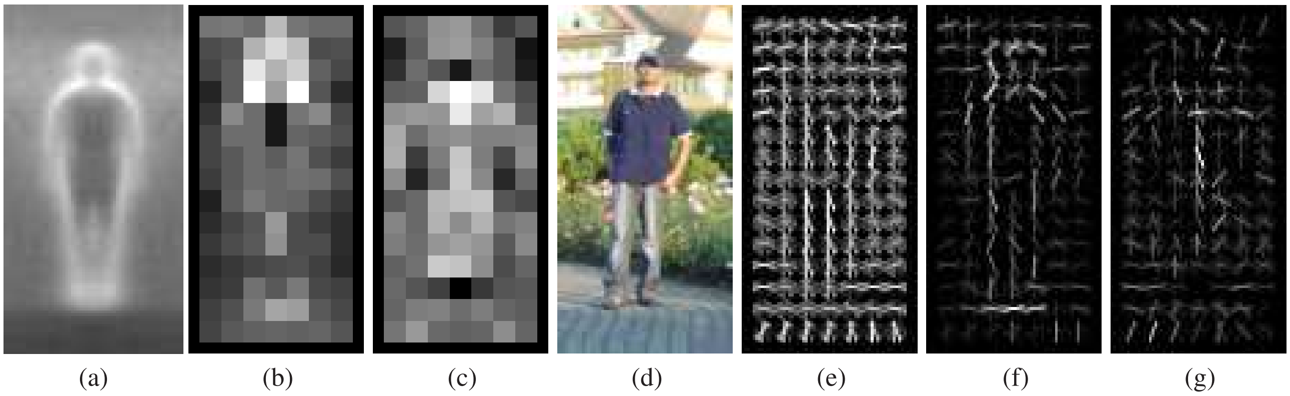


Figure 6. Our HOG detectors cue mainly on silhouette contours (especially the head, shoulders and feet). The most active blocks are centred on the image background just *outside* the contour. (a) The average gradient image over the training examples. (b) Each "pixel" shows the maximum positive SVM weight in the block centred on the pixel. (c) Likewise for the negative SVM weights. (d) A test image. (e) It's computed R-HOG descriptor. (f,g) The R-HOG descriptor weighted by respectively the positive and the negative SVM weights.

results are insensitive to ϵ's value over a large range.

Centre-surround normalization. We also investigated an alternative centre-surround style cell normalization scheme, in which the image is tiled with a grid of cells and for each cell the total energy in the cell and its surrounding region (summed over orientations and pooled using Gaussian weighting) is used to normalize the cell. However as fig. 4(c) ("*window norm*") shows, this decreases performance relative to the corresponding block based scheme (by 2% at 10^{-4} FPPW, for pooling with $\sigma{=}1$ cell widths). One reason is that there are no longer any overlapping blocks so each cell is coded only once in the final descriptor. Including several normalizations for each cell based on different pooling scales σ provides no perceptible change in performance, so it seems that it is the existence of several pooling regions with *different* spatial offsets relative to the cell that is important here, not the pooling scale.

To clarify this point, consider the R-HOG detector with overlapping blocks. The coefficients of the trained linear SVM give a measure of how much weight each cell of each block can have in the final discrimination decision. Close examination of fig. 6(b,f) shows that the most important cells are the ones that typically contain major human contours (especially the head and shoulders and the feet), normalized w.r.t. blocks lying *outside* the contour. In other words — despite the complex, cluttered backgrounds that are common in our training set — the detector cues mainly on the contrast of silhouette contours against the background, not on internal edges or on silhouette contours against the foreground. Patterned clothing and pose variations may make internal regions unreliable as cues, or foreground-to-contour transitions may be confused by smooth shading and shadowing effects. Similarly, fig. 6(c,g) illustrate that gradients inside the person (especially vertical ones) typically count as negative cues, presumably because this suppresses false positives in which long vertical lines trigger vertical head and leg cells.

6.5 Detector Window and Context

Our $64{\times}128$ detection window includes about 16 pixels of margin around the person on all four sides. Fig. 4(e) shows that this border provides a significant amount of context that helps detection. Decreasing it from 16 to 8 pixels ($48{\times}112$ detection window) decreases performance by 4% at 10^{-4} FPPW. Keeping a $64{\times}128$ window but increasing the person size within it (again decreasing the border) causes a similar loss of performance, even though the resolution of the person is actually increased.

6.6 Classifier

By default we use a soft ($C{=}0.01$) linear SVM trained with SVMLight [10] (slightly modified to reduce memory usage for problems with large dense descriptor vectors). Using a Gaussian kernel SVM increases performance by about 3% at 10^{-4} FPPW at the cost of a much higher run time.

6.7 Discussion

Overall, there are several notable findings in this work. The fact that HOG greatly out-performs wavelets and that any significant degree of smoothing before calculating gradients damages the HOG results emphasizes that much of the available image information is from *abrupt edges at fine scales*, and that blurring this in the hope of reducing the sensitivity to spatial position is a mistake. Instead, gradients should be calculated at the finest available scale in the current pyramid layer, rectified or used for orientation voting, and only then blurred spatially. Given this, relatively coarse spatial quantization suffices (6–8 pixel wide cells / one limb width). On the other hand, at least for human detection, it pays to sample orientation rather finely: both wavelets and shape contexts lose out significantly here.

Secondly, strong *local* contrast normalization is essential for good results, and traditional centre-surround style schemes are not the best choice. Better results can be achieved by normalizing each element (edge, cell) *several times* with respect to different local supports, and treating the results as independent signals. In our standard detector, each HOG cell appears four times with different normalizations and including this 'redundant' information improves performance from 84% to 89% at 10^{-4} FPPW.

7 Summary and Conclusions

We have shown that using locally normalized histogram of gradient orientations features similar to SIFT descriptors [12] in a dense overlapping grid gives very good results for person detection, reducing false positive rates by more than an order of magnitude relative to the best Haar wavelet based detector from [17]. We studied the influence of various descriptor parameters and concluded that fine-scale gradients, fine orientation binning, relatively coarse spatial binning, and high-quality local contrast normalization in overlapping descriptor blocks are all important for good performance. We also introduced a new and more challenging pedestrian database, which is publicly available.

Future work: Although our current linear SVM detector is reasonably efficient – processing a 320×240 scale-space image (4000 detection windows) in less than a second – there is still room for optimization and to further speed up detections it would be useful to develop a coarse-to-fine or rejection-chain style detector based on HOG descriptors. We are also working on HOG-based detectors that incorporate motion information using block matching or optical flow fields. Finally, although the current fixed-template-style detector has proven difficult to beat for fully visible pedestrians, humans are highly articulated and we believe that including a parts based model with a greater degree of local spatial invariance would help to improve the detection results in more general situations.

Acknowledgments. This work was supported by the European Union research projects ACEMEDIA and PASCAL. We thanks Cordelia Schmid for many useful comments. SVM-Light [10] provided reliable training of large-scale SVM's.

References

[1] S. Belongie, J. Malik, and J. Puzicha. Matching shapes. *The 8th ICCV, Vancouver, Canada*, pages 454–461, 2001.

[2] V. de Poortere, J. Cant, B. Van den Bosch, J. de Prins, F. Fransens, and L. Van Gool. Efficient pedestrian detection: a test case for svm based categorization. *Workshop on Cognitive Vision*, 2002. Available online: http://www.vision.ethz.ch/cogvis02/.

[3] P. Felzenszwalb and D. Huttenlocher. Efficient matching of pictorial structures. *CVPR, Hilton Head Island, South Carolina, USA*, pages 66–75, 2000.

[4] W. T. Freeman and M. Roth. Orientation histograms for hand gesture recognition. *Intl. Workshop on Automatic Face- and Gesture- Recognition, IEEE Computer Society, Zurich, Switzerland*, pages 296–301, June 1995.

[5] W. T. Freeman, K. Tanaka, J. Ohta, and K. Kyuma. Computer vision for computer games. *2nd International Conference on Automatic Face and Gesture Recognition, Killington, VT, USA*, pages 100–105, October 1996.

[6] D. M. Gavrila. The visual analysis of human movement: A survey. *CVIU*, 73(1):82–98, 1999.

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

[8] D. M. Gavrila and V. Philomin. Real-time object detection for smart vehicles. *CVPR, Fort Collins, Colorado, USA*, pages 87–93, 1999.

[9] S. Ioffe and D. A. Forsyth. Probabilistic methods for finding people. *IJCV*, 43(1):45–68, 2001.

[10] 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.

[11] Y. Ke and R. Sukthankar. Pca-sift: A more distinctive representation for local image descriptors. *CVPR, Washington, DC, USA*, pages 66–75, 2004.

[12] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.

[13] R. K. McConnell. Method of and apparatus for pattern recognition, January 1986. U.S. Patent No. 4,567,610.

[14] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *PAMI*, 2004. Accepted.

[15] K. Mikolajczyk and C. Schmid. Scale and affine invariant interest point detectors. *IJCV*, 60(1):63–86, 2004.

[16] K. Mikolajczyk, C. Schmid, and A. Zisserman. Human detection based on a probabilistic assembly of robust part detectors. *The 8th ECCV, Prague, Czech Republic*, volume I, pages 69–81, 2004.

[17] A. Mohan, C. Papageorgiou, and T. Poggio. Example-based object detection in images by components. *PAMI*, 23(4):349–361, April 2001.

[18] C. Papageorgiou and T. Poggio. A trainable system for object detection. *IJCV*, 38(1):15–33, 2000.

[19] R. Ronfard, C. Schmid, and B. Triggs. Learning to parse pictures of people. *The 7th ECCV, Copenhagen, Denmark*, volume IV, pages 700–714, 2002.

[20] Henry Schneiderman and Takeo Kanade. Object detection using the statistics of parts. *IJCV*, 56(3):151–177, 2004.

[21] Eric L. Schwartz. Spatial mapping in the primate sensory projection: analytic structure and relevance to perception. *Biological Cybernetics*, 25(4):181–194, 1977.

[22] P. Viola, M. J. Jones, and D. Snow. Detecting pedestrians using patterns of motion and appearance. *The 9th ICCV, Nice, France*, volume 1, pages 734–741, 2003.