Real-Time Humans Detection in Urban Scenes

Julien Bégard, Nicolas Allezard and Patrick Sayd
CEA, LIST,
Boîte Courrier 65, Gif sur Yvette, F-91191 France
(julien.begard,nicolas.allezard,patrick.sayd)@cea.fr

Abstract

We address the issue of real-time pedestrians detection in a urban environment. This is a challenging task owing to the high variability of appearances and poses that humans can have and to the complexity of backgrounds. We propose a solution made of gradient-based local descriptors combined to form strong classifiers and organized in a cascaded detector. We developed for this an extension of the Histograms of Oriented Gradients (HOGs) and added a new component to the histogram which represents the *strength* of edges or the *amount of information* in the histogram support. We also implemented a learning algorithm based on Real Adaboost where two phases – selection first, then refinement of weights – provide more robustness to the detector. We evaluated our system by comparing it to the cascaded detector of Haar features of Viola & Jones [7] and to the SVM of HOGs features of Dalal & Triggs [1]. To ensure an equitable and valid comparison, we used the database proposed in [1]. Our system outperforms them in detection results and in time needs.

1 Introduction

The human class in object detection is probably one of the more difficult because of the variability of appearances and poses that humans can have. The descriptor used to describe and to detect human silhouettes has to capture finely the good characteristics to assure an efficient detection. Moreover, applied to an urban context, the detection chain has to be as robust as possible and has to run in real-time. We developed a human detection system based on Histograms of Oriented Gradients (HOGs) features learnt by a cascaded boosting procedure which performs better than Viola-Jones [7] Haar+Adaboost cascade (**system 1**) and Dalal-Triggs [1] HOG+SVM (**system 2**). We adopted the cascade approach from [7] with some modifications which tend to make the system more robust and we worked with HOGs-like descriptors including gradient strength information. This paper is organized as follow: we first make an overview of previous work on human detection in 2. Then we present briefly our system in 3, experimental results are exposed in 4 with a comparison with other state-of-the-art methods. In 5, we present the performance of our system through several parameters study and we finally conclude and discuss about future work in 6.

2 Previous Work

Lot of work has already been done in Computer Vision based detection systems and an extensive literature accompanies this work. Papageorgiou *et al* [6] presented a detector which infers a pedestrian model from positive and negative examples by the mean of a polynomial SVM and uses *quadruple density* Haar Wavelets as a pattern descriptor. An improved version has been developed by Deporrtere *et al* [2], principally with dimension reduction thanks to Adaboost features selection and SVRM[1] model generation. Viola *et al* [7] introduced a Haar-like wavelets coarse-to-fine cascaded face detector combined with an efficient features extraction method – the integral image – and extended this system to take into account motion cues in video scenes [8]. The system developed by Gravila *et al* [3] computes the chamfer distance to perform a shape-based pedestrian detection and validates the detection with textures classification from a neural network and stereo verifications. Mikolajczyk *et al* [5] assembled seven part detectors (frontal and profile head, face and upper body, legs) with Bayesian decision rules. Dalal and Triggs [1] proposed a detector build on a SVM learning machine and Histograms of Oriented Gradients (HOG) through a very interesting study of their implementation issues. A more challenging goal is aimed by Leibe *et al* [4] and Wu and Nevatia [9] who tried to detect partially occluded pedestrians in crowded scenes. The formers considered the aggregation of local and global cues while the latter used *edgelet* features learnt by a boosting method.

3 Overview of the System

In this section, we describe briefly our system schematized in figure 1. More details for each parameters are given in section 5.

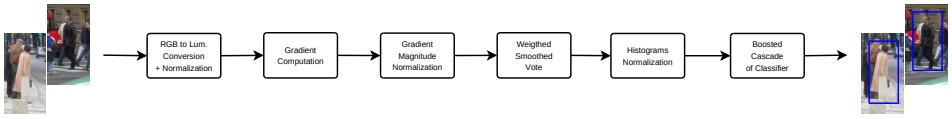

Figure 1: Detection chain of our system.

We developed a system that can achieve real-time pedestrians detection in urban scenes with very low false-positive rates. Real-time detection is an important point for this system which could be adapted to a future embedded version.
Our method is based on a true/false classification obtained by an evaluation of local descriptors computed densely on input images through a boosted cascade of classifiers.

3.1 Local Descriptors

Distinction and recognition of objects in static scenes can be obtained from shape contours. HOGs work well to capture such information from an image but it lacks of a clue, which could be named the *contour strength*, to make difference between acute and loose edges. To correct this drawback, we included the gradient magnitude as a new component in the histogram so that our histograms are now made of 9 magnitude-normalized orientations bins – each one 20°wide – plus 1 bin for the gradient magnitude. The orientations

[1] Support Vector Regression Machines

are unsigned so that we do not make difference between a dark-bright transition and a bright-dark one (this assumption makes sense with the high color and texture variability of human appearances: hair, skin, clothes, etc.). To reduce aliasing, we use a kind of simple smoothing to compute the histogram components by giving a fraction x of the vote to the corresponding bin and a fraction $1-x$ to the nearest bin where $x \in [x_{min}, 1]$. x_{min} depends on the angle threshold θ_T above which we consider a vote belonging to one and only one bin ($x = 1$). In fine, we obtain heterogeneous descriptors, computed in a dense way, that can code the orientations and the *strength* of a shape.
Input images are represented in grayscale color space and they are first normalized along the luminance in areas where there are enough gradient information. This preprocessing provides to the system some robustness to illumination conditions. Gradients are computed with an optimized implementation of the Deriche operator and components votes are efficiently accumulated thanks to the integral image techniques.

3.2 Learning Procedure

Pedestrians characteristics are extracted and learnt by a cascaded boosting algorithm like the one used in [7]. In our case the learning algorithm has to face with a very high dimensional space made of almost ten thousand of components for each positive and negative example. The Adaboost algorithm is used to select the relevant components and to train classifiers from them like **system 1**. The boosting algorithm builds, for each stage, a strong classifier from a weighted selection of features – the weak classifiers – and stop selecting features when the training errors drop below a threshold.
We changed a bit this procedure to robustify the resultant strong classifier: we first select n weak classifiers, each of them is a component of an histogram. Viola's algorithm would stop there but ours comprises another loop of m rounds of boosting to compute more finely the weights of the n selected components. The difficulty is to find the two optimal parameters n and m to prevent the system from over-learning, which could lead to a decrease in the performances.

4 Results and Comparison with Different Systems

4.1 Databases

Our system is dedicated to human detection in urban scenes. To train and test it, we acquired several hours of video with a video-camera fixed on a car driven in cities (see fig. 2 for examples of image). We obtained very good results on this database. Although we could use these datas in this paper, we would prefer using the database available in [1], where the background is not especially urban, to make more neutral comparisons between the different systems. This database contains upright humans with various poses, clothes, backgrounds and light conditions. Some samples have partial occlusions. There are 2478 96×160 positives examples for training (1239 + left-right reflexions) and 1126 70×134 positives examples (563 + reflexions) for testing. People in these images are 64×128 sized. There is also a free-person set of 1218 images for generating the negative examples.

4.2 Methodology.

We trained and tested our detector on this database in two different ways. The system presented in [7] is based on a cascade of boosted classifiers whereas the system in [1]

Figure 2: Samples of images in our urban database.

Figure 3: Some samples from the database we used.

is based on a linear SVM. Although the cascade hierarchy provides a faster behavior to the system, it also decreases a little the detection rate. Indeed, several stages have more chance to reject wrongly good candidates than only one stage. This is why we separated our comparisons in two distinct parts: the first comparison will be between the cascades of classifiers, a Haar-based from Viola-Jones and our cascade. The second comparison will compare the SVM+HOGs system from Dalal-Triggs with a *one stage* cascade trained with our method. For each case, we give ROC curves with Miss Rate ($1-$Recall) versus False Positive Per Windows (FPPW) rate.

4.3 Results & Comparisons.

Boosted cascade of Haar classifiers vs. improved boosted cascade of HOGs. We used Intel OpenCV implementation of Viola-Jones detector and we trained it with the datas we described above. We gave the same parameters to the two training programs: 2478 positives samples and 10000 negatives samples and we built two cascades made of 10 stages with the Real version of Adaboost. Minimum hit rate and maximum false alarm rate are respectively fixed to 99.5% and 50%. Training has been done on an Intel Xeon Processor at 3.0GHz equiped with 4Gb of memory and it took several days (almost 1 week) for **system 1** whereas ours needed only a few hours (almost 2 hours). The final cascades used 397 different weak classifiers for **system 1** and 169 for ours.
Now let's talk about results and performance. Figure 4 clearly shows that our cascaded pedestrian detector is better than **system 1**. Further studies shows that **system 1** has difficulties to detect correctly pedestrians present in the database we used. This is due to the variety of backgrounds and parasitic elements like traffic signs, billboards, etc. By the same way, those same elements add false alarms leading to an overall decrease of the performance. Considering processing times, it took 46.4 ms for **system 1** and 27 ms for our system to scan a 320×240 image with the same parameters (scale factor of 1.2, step size of 1.5). The figure 4 shows the ROC curves for **system 1** and our system.

HOGs+linear SVM vs. *1-stage* HOGs+Boosting. We used the latest system avail-

able described in [1] to compare it with our method. We let the default parameters as there are supposed to be the best for this system and we trained it with the same examples set, which took several hours (almost 3). To have a comparable system, we trained a *one stage cascade* with our own program with 50 weak classifiers. This was done in less than an hour.
Regarding the results, our system quickly outperforms the system based on HOGs+linear SVM and provides a near perfect detection with an acceptable false positive rate. Our system is a little bit less performant for FPPW under $5E-5$ but above, it is better than **system 2** and reaches perfect detection for false alarm rates after $6E-4$. See fig. 4 for the whole ROC curves. We also compared processing time on 320×240 images with a dense scan of almost 5500 detection windows: **system 2** took 3 seconds whereas ours needed only 229ms. Remind that the cascaded detector needed only 27ms: this shows how the cascade structure is efficient to reduce detection times.

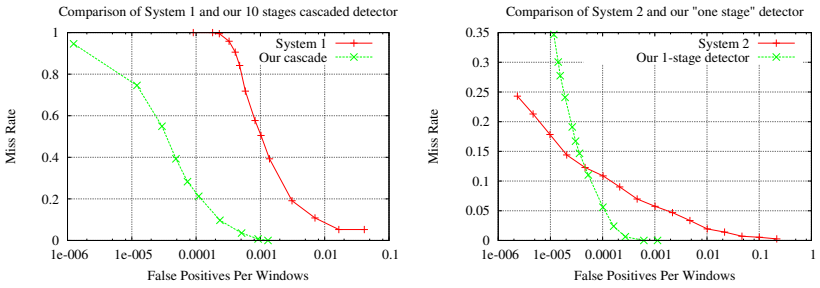

Figure 4: Performances of **system 1**and our cascade (left) and performances of **system 2**and our *1-stage* detector (right). In both cases, our method outperforms the other systems.

Figure 11 shows examples of output of our systems on some test images of the database from [1].

5 Performance and Parameters Study

In this section, we study the impact of some parameters on the system. After addressing the issue of gradient computation method, we justify the use of the gradient *strength* as a new component in the histogram and then study the drawing of the dense grid on which histograms are computed. We expose then the initialization of the learning algorithm and the manner in which a strong classifier is trained. After that, we discuss of the existence of an optimal number of stage in a cascaded detector and we finally study the structure and characteristics of the resultant detector.

5.1 Gradient Computation

The detection chain almost begins with gradient computation so that the detector results are closely dependent with the way in which they are computed. We tested several gradient computation methods to select the best result/time ratio as we want our system to be real-time. Rapid gradient computations could be done with simple small mask derivation (1D: $[-1,0,1]$, 2D: Sobel) and also with the Deriche recursive operators.

The only parameter for Deriche's operators is α which drives the *smoothing* phase of the filter. The smaller α is, the stronger the smoothing will be. But, as our descriptor finds complementary information on the gradient strength, α must not be too small.
We looked then at the influence of the size of the normalization windows. As expected, the evolution of the results reached an optimum as the size increases. We started from 10 pixels to 30 pixels and the optimum is between 15 and 20 pixels. Note that this window is as broad as an arm/leg of human on the training samples.

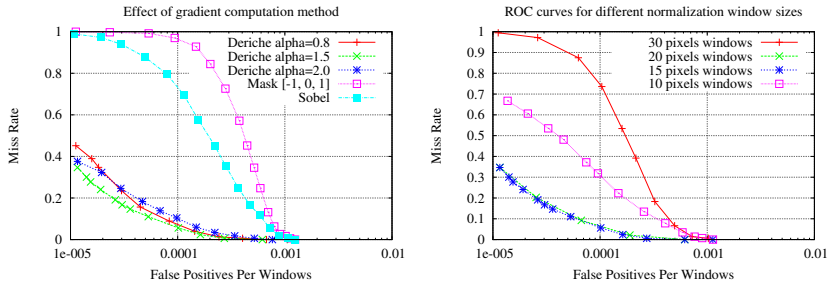

Figure 5: Effect of gradient computation method (left) and of normalization window size (right).

5.2 Contribution of the Gradient Magnitude in the Descriptor

We now study the contribution of the gradient magnitude that we called the *contour strength*. This is done by comparing two detector: one with classic HOG descriptors and the other with HOG+magnitude. The two systems differ only on the descriptors used, they are trained with the same parameters and on the same database. Figure 6 clearly shows that including the magnitude of the gradient as a new component of the histogram improves the results.

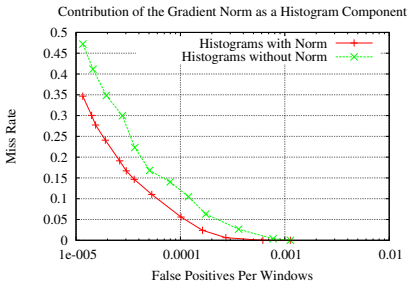

Figure 6: Comparison between HOGs with norm and without norm. There is a positive gain of almost 0.1 between the two descriptors at no additional calculation cost.

5.3 Drawing a Dense Grid on the Image

Our system is sensitive to the gradients computation method, but obviously it is also sensitive to the manner used to *draw* a dense grid on inputs image. We evaluated different

schemes with fixed sizes of windows (8×8, 16×16, 24×24) and with relative sizes of windows ($[0.1 \rightarrow 0.9] \times$ *ImgSize* with position steps of 0.1 and a scale factor steps of 0.2 for example). Our tests arrive at the conclusion that a multiscale grid is better and that it is preferable to let the learning algorithm run in a very high dimensional space than reducing this dimension by fixing arbitrarily restrictive sizes for the computation grid. Figure 7 shows this results.

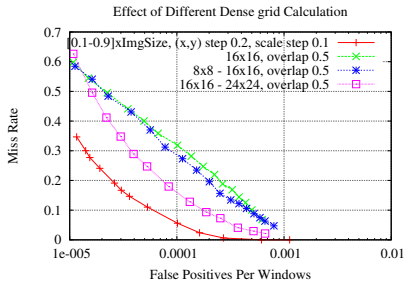

Figure 7: The results of the detector depends on how the dense grid of histograms is computed. A multiscale grid is better and the results are even more better if the histograms are computed on regions of the size of a human limb.

5.4 The Cascaded Detector

We noticed several important points as we worked on our cascaded pedestrian detector. First is the issue of the initialization of the learning algorithm. There are two possibilities to initialize the weights of the training samples: 1. $w_+ = w_- = \frac{1}{N_+ + N_-}$ or 2. $w_+ = \frac{1}{N_+}$ *and* $w_- = \frac{1}{N_-}$ with $N_{+/-}$ the number of positive/negative examples. The second initialization method leads in fact to two cases depending on the amount of positive and negative examples. We could intuitively think that giving a better weights to positive (or negative) examples will influence the system so that it focuses its discriminative work on them. This is true but finally, this does not improve the results at all. It is better to let the system find his own rules with an equitable initialization.
The second point concerns the number of stages of the cascade. As we know, the cascade structure is useful to accelerate the processing as it uses a coarse-to-fine approach. Although this technique degrades a little the detection results, the gain in performance is satisfactory enough to justify this loss. Nevertheless, the cascade length has a limit beyond which the system speed stagnates whereas the results decrease too much. Thus, this indicates that there would be an optimal number of stages for building a cascaded detector. We verified this and tried to show the independence of this number of stages to the testing set. Those two points are illustrated by figure 8.

5.5 Training a Strong Classifier

The learning algorithm we developed for our system is quite similar as the one used in **system 1**. But, whereas **system 1** lets the Adaboost algorithm run until it found the required features, we force our algorithm – a classical Real Adaboost – to begin another loop of rounds of boosting after the selection phase. The idea is to first select the best

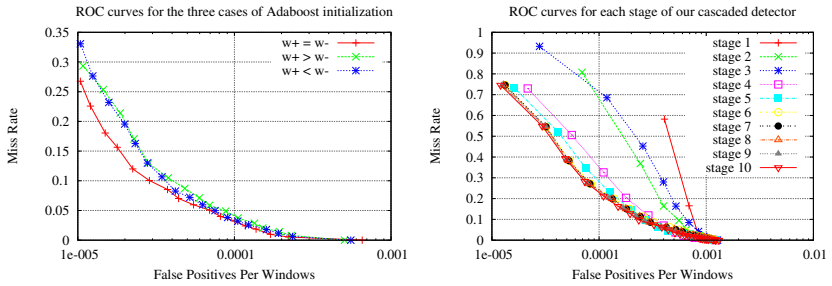

Figure 8: Left: effect of the initialization of the learning algorithm. Right: The ROC curves for each stage of our cascaded detector. The results quickly reach an asymptote.

features and then to refine the weights as precisely as possible.
For this, we have to determine previously the number of descriptors and also the number of additional rounds of boosting to calculate the weights. So we study the influence of those two parameters on our system whose results reach an asymptote as they increase. We noticed that a strong classifier built with very few descriptors is very fast, of course, but useless because it is not discriminant enough. On the other hand, a big strong classifier has good results but is also very slow. As an example, we built our *one stage* detector with 50 descriptors boosted by 10 rounds/descriptors.

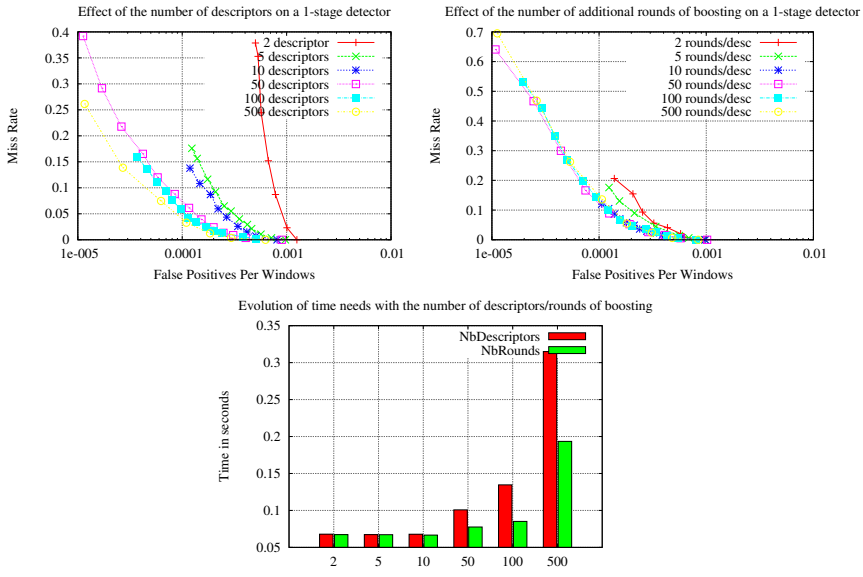

Figure 9: Effect of the number of descriptors and the number of additional rounds of boosting on a 1-stage detector. We choose, as a compromise between detection rates and time needs, 50 descriptors for 10 rounds per desc. of additional boosting iterations.

5.6 Post Learn Study

Results and detection speed are good indicators to evaluate our system, but we would like to go further and to see how our learning machine built the detector. We observed that *stronger* histograms are chosen more often than others. In training samples, this corresponds to select the edges of the silhouettes we want to learn (see fig. 10). Moreover, when we focus on descriptors that are the most chosen, or on descriptors that are the most discriminant – they have the strongest positive/negative vote – we noticed that they are localized in regions like *head*, *shoulders*, *legs* and the *crotch*. That is to say characteristic regions that allow to recognize a pedestrian in an image. Finally, histograms from regions as broad as a human limb are also selected more often than others.

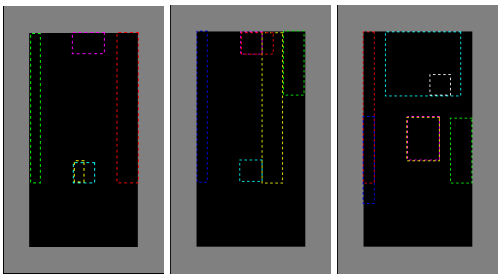

Figure 10: Examples of support of HOG selections. The main regions are near the head, shoulders, legs and also the crotch. Pedestrians are in 64×128 bounding boxes (black) inside 96×160 images (gray).

6 Conclusion and Future Work

We have proposed a system that detects pedestrians – vulnerable users of the road – in real time. To build this system, we used HOGs local descriptors, computed on a dense overlapping grid and learnt by a boosting algorithm whose output is a cascaded detector. We have shown that considering the gradient magnitude as a new component in the HOGs provides new information to the learning algorithm and improves dramatically the results. We have also presented a learning procedure based on Adaboost made of two phases – a feature selection phase followed by the precise refinement of the weights – which adds robustness to the system while improving the detection rates. We evaluated our system and compared it to Viola-Jones one and Dalal-Triggs one in equitable conditions and showed that our detector performs the best.

Although detection rates and speeds are satisfactory, there is obviously still room for optimizations. The computation of the score of the strong classifiers is far from being efficient since we did not limit the displacements in the image. In the same way, the boosting algorithm we used is an accumulating algorithm. That is to say the only action it does is to add features to the classifier and adjust the weights. We could employ another algorithm that, for every iteration, decides whether to add a new feature or remove an old feature. Context in the image also provides a lot of information and could be used, for example, to remove false alarms or to get rid of some partial occlusions. Finally, we currently begin to study how to combine the histograms together to reinforce the selection of the learning algorithm. The idea beyond this is intuitive: an histogram is selected

because it contains discriminative information. So its direct neighbors could also contain this information, even in a lower quantity. We call this *Adjacent HOGs*.

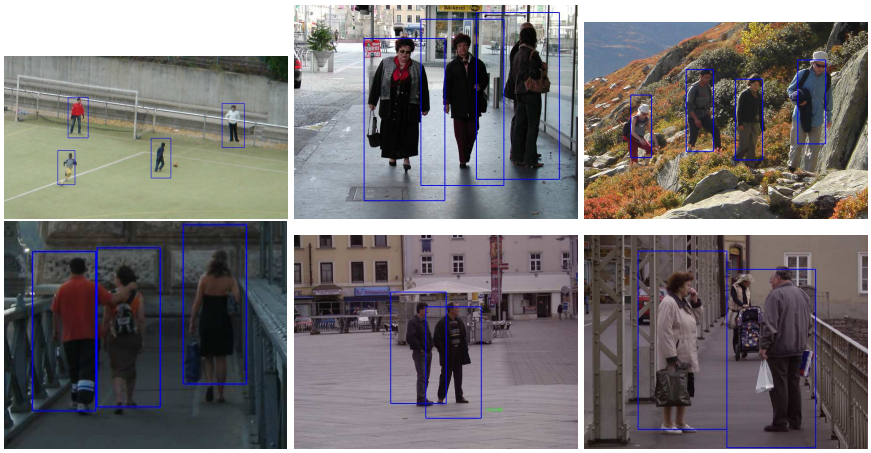

Figure 11: Some examples of detection on full size images.

References

[1] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, volume II, pages 886–893, 2005.

[2] V. Deporrtere, J. Cant, B. Van den Bosh, J. De Prins, R. Fransens, and L. Van Gool. Efficient pedestrian detection: a test case for svm based categorization. In *Workshop on Cognitive Vision*, 2002.

[3] D.M. Gravila, Giebel J., and Munder S. Vison-based pedestrian detection: The protector system. In *IEEE Intelligent Vehicles Symposium*, 2004.

[4] B. Leibe, E. Seemann, and B. Schiele. Pedestrian detection in crowded scenes. In *CVPR*, volume I, pages 878–885, 2005.

[5] K. Mikolajczyk, C. Schmid, and A. Zisserman. Human detection based on a probabilistic assembly of robust part detectors. In *ECCV*, volume I, pages 69–81, 2004.

[6] C. Papageorgiou, T. Evgeniou, and T. Poggio. A trainable pedestrian detection system. In *IEEE Intelligent Vehicles Symposium*, pages 241–246, 1998.

[7] P. Viola and M. Jones. Robust real-time object detection. In *International Journal of Computer Vision*, 2002.

[8] P. Viola, M. Jones, and D. Snow. Detecting pedestrians using patterns of motion and appearance. In *ICCV*, 2003.

[9] B. Wu and R. Neviata. Detection of multiple, partially occluded humans in a single image by bayesian combination of edgelet part detectors. In *ICCV*, pages 90–97, 2005.