Real-time Pedestrian Detection with Deformable Part Models

Hyunggi Cho, Paul E. Rybski, Aharon Bar-Hillel and Wende Zhang

***Abstract*— We describe a real-time pedestrian detection system intended for use in automotive applications. Our system demonstrates superior detection performance when compared to many state-of-the-art detectors and is able to run at a speed of 14 fps on an Intel Core i7 computer when applied to 640×480 images. Our approach uses an analysis of geometric constraints to efficiently search feature pyramids and increases detection accuracy by using a multiresolution representation of a pedestrian model to detect small pixel-sized pedestrians normally missed by a single representation approach. We have evaluated our system on the Caltech Pedestrian benchmark which is currently the largest publicly available pedestrian dataset at the time of this publication. Our system shows a detection rate of 61% with 1 false positive per image (FPPI) whereas recent other state-of-the-art detectors show a detection rate of 50% ∼ 61% under the *'reasonable'* test scenario (explained later). Furthermore, we also demonstrate the practicality of our system by conducting a series of use case experiments on selected videos of Caltech dataset.**

I. INTRODUCTION

Vision-based pedestrian detection is a popular research topic in the computer vision community due to direct application of topics such as visual surveillance [20] and automotive safety [18], [14]. In the past few years, impressive progress has been made, such as found the works of [17], [19], [4], [11], [24], [6], and these insights suggest that they can and should be applied to real world applications. Within the past few years, extremely challenging real-world datasets such as the Caltech Pedestrian [7] and Daimler Pedestrian [8] collections have been introduced. These public datasets allow researchers to evaluate their algorithm's performance against that of other researchers. We have been making use of these in our research.

Carnegie Mellon University won the 2007 DARPA Urban Challenge with the autonomous vehicle "Boss" [9]. However, in that race, no pedestrians were allowed on the track. While impressive, the technology necessary to win that race is insufficient for operating on real roads. Our recent work has been to focus primarily on developing a real-time pedestrian detection system for automotive safety applications which could be used by both an autonomous vehicle to help it safely navigate roads as well as by a manually-driven vehicle as an early-warning system for the driver. The contributions of this work are threefold.

H. Cho and P. E. Rybski are with ECE and the Robotics Institute, respectively, Carnegie Mellon University, 5000, Forbes Ave., Pittsburgh, PA 15213, USA. {hyunggic, prybski}@cs.cmu.edu

A. Bar-Hillel is with the Advanced Technical Center, General Motors, Hamanofim 11, Herzliya, Israel. aharon.barhillel@gm.com

W. Zhang is with the Electrical and Controls Integration Lab, General Motors R&D, 30500, Mound Rd, Warren, MI 48092, USA. wende.zhang@gm.com

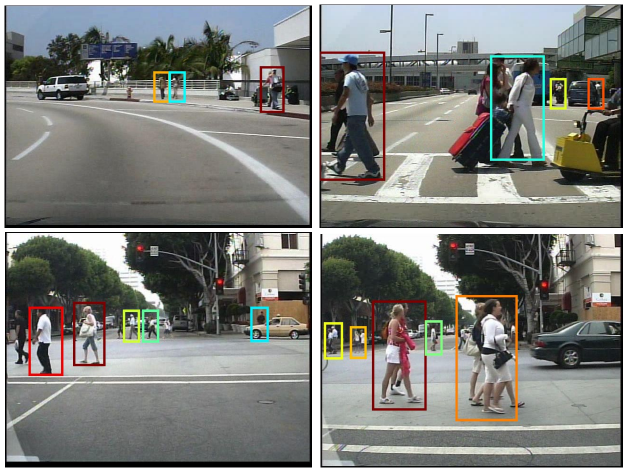


Fig. 1. Typical detection results of our on-board pedestrian detection system. With a multiresolution deformable part-based model, the system can detect pedestrians up to 25m reliably.

The first main contribution is a C implementation of our detection system suitable for real-time operation. Achieving real-time detection speed is by no means a trivial process especially for most of the state-of-the-art detectors which usually use a sliding window method. Our detection system is based on the star-cascade algorithm [10] for a part-based deformable model [11] introduced by Felzenszwalb et al. which is considered as one of the most successful approaches in general object detection. Because of the generality of this method, we can relatively easily apply our system to other relevant categories such as vehicles and bicycles by providing a new training set. In our implementation, we simplify the structure of the star-cascade model for improved speedup and demonstrate a real-time performance with a vehicle-mounted camera.

The second main contribution is a safe geometric constraint analysis based on known camera calibration information for efficient search. Usually, the availability of such information is not assumed in general object detection applications, but for automotive applications where the camera will be mounted in a known position in the vehicle, this information is very useful to exploit. By doing so, we are able to not only accelerate our detection process by searching only relevant image spaces, but we are also able to improve detection accuracy by suppressing a number of potential false positives from irrelevant image space. Based on some assumptions, we analyze the following relationships: 'pedestrian height in pixels' vs. 'distance from a vehicle' and 'pedestrians' foot position in images' vs. 'distance from a vehicle'. We propose a simple algorithm for fast search based on these relationship.

Our third main contribution is a quantitative evaluation of our system using public real world datasets. First, we compared our detector's performance with current state-of-the-art detectors using the Caltech Pedestrian Benchmark. The Caltech dataset is at least two orders of magnitude larger than most exting datasets and provides us a unique and interesting opportunity for in-depth system evaluation. Through a series of well designed experiments, we seek to identify values for key design parameters of our detector such as the optimal number of parts for our deformable part-based model, the optimal number of scales per octave for multiscale detection, etc.

The remainder of this paper is organized as follows. Section II reviews related work on pedestrian detection. Technical implementation details of our detection system are described in Section III and a geometric constraint from known camera calibration information is exploited for an efficient search in Section IV. We describe experimental results using the system in Section V and conclude in Section VI.

II. RELATED WORK

For a comprehensive survey of recent works in vision-based pedestrian detection, refer to [7], [15]. Dollár et al. [7] focuses primarily on the pedestrian detection problem and performs an intensive evaluation of the majority of the state-of-the-art detector algorithms. Gerónimo et al. [15] focuses on pedestrian protection systems for advanced driver assistance systems which utilize tracking, scene geometry, and stereo systems. We review here only important advances for pedestrian detection (not tracking) and describe how these detection approaches can be specially designed for automotive applications.

Historically, one of the first pioneering efforts was the work of Papageorgiou et al. [17] which used Haar wavelet features in combination with a polynomial Support Vector Machine (SVM). They also introduced the first generation pedestrian dataset, known as the 'MIT Pedestrian Dataset'. Inspired by their work, Viola and Jones [19] brought several important ideas into this field including the use of a new machine learning algorithm (AdaBoost) for automatic feature selection, the use of a cascade structure classifier for efficient detection, and finally the use of an integral image concept for a fast feature computation. Later, the authors demonstrated how to incorporate space-time information into their simple Haar-like wavelet features for moving people detection [20].

The next breakthrough in the detection technology occured in a feature domain itself for pedestrian detection. Dalal and Trigg [4] showed excellent performance for detecting a human in a static image using dense HOG (Histogram of Oriented Gradient) features with linear SVM. They also introduced the second generation pedestrian dataset, called the 'INRIA Person Dataset.' Currently, HOG is considered to be the most discriminative single feature and is used in nearly all modern detectors in some forms [7].

Detectors that improve upon the performance of HOG utilize a fusion of multiple features and part-based approaches. Wojek and Schiele [23] exploit several combinations of multiple features such as Haar-like features, shapelets, shape context, and HOG features. This approach is extended by Walk et al. [21] by adding local color self-similarity and motion features. Wang et al. [22] combined a texture descriptor based on local binary patterns with HOG. Recently, Dollár et al. [6] provided a simple and uniform framework for integrating multiple feature types including LUV color channels, grayscale, and gradient magnitude quantized by orientation. That group also implemented a near real-time version of this algorithm [5] which makes this method suitable for automotive applications.

Part-based approachs have gained popularity recently mainly because they can handle the various appearances of pedestrians (due to clothing, pose, and occlusion) and can provide a more complex model for pedestrian detection. Mohan et al. [17] use this methodology to divide the human body into four parts: head, legs, left arm, and right arm. Each part detector is trained using a polynomial SVM where outputs are fed into a final classifier after checking geometric plausibility. Mikolajczyk et al. [16] model humans as assemblies of parts that are represented by SIFT-like orientation features. Felzenszwalb et al. [11] demonstrated that their deformable part-based model human detector can outperform many of existing current single-template-based detectors [20], [4], [22]. Based on a variation of HOG features, they introduce a latent SVM formulation for training a part-based model from overall bounding box information without part location labels. Bar-Hillel et al. [1] introduced a new approach for learning part-based human detection through feature synthesis.

Pedestrian detection algorithms have an obvious extension to automotive applications due to the potential for improving safety systems. In this case, the design criterion for a detector might be very different as real-time operation is just important as high detection accuracy. Shashua at al. [18] proposed a part-based representation in a fixed configuration for pedestrian detection. The authors used 13 overlapping parts with HOG-like features and ridge regression to learn a classifier for each part and reported a classification performance of a 93.5% detection rate and a 8% false positive rate. Gavrila and Munder [14] proposed a pipline using Chamfer matching and several image based verification steps for a stereo camera setup. The classification performance was reported as a 90% detection rate with a 10% false positive rate.

Our group has been making use of the deformable part-based model [11] as part of a tracking-by-detection system that is intended for use with an autonomous vehicle as well as an early-warning system for driver safety. We have demonstrated the algorithmic viability of these methods in off-line analysis [2], [3] and in this work we seek to demonstrate how these approaches can be implemented in real-time.

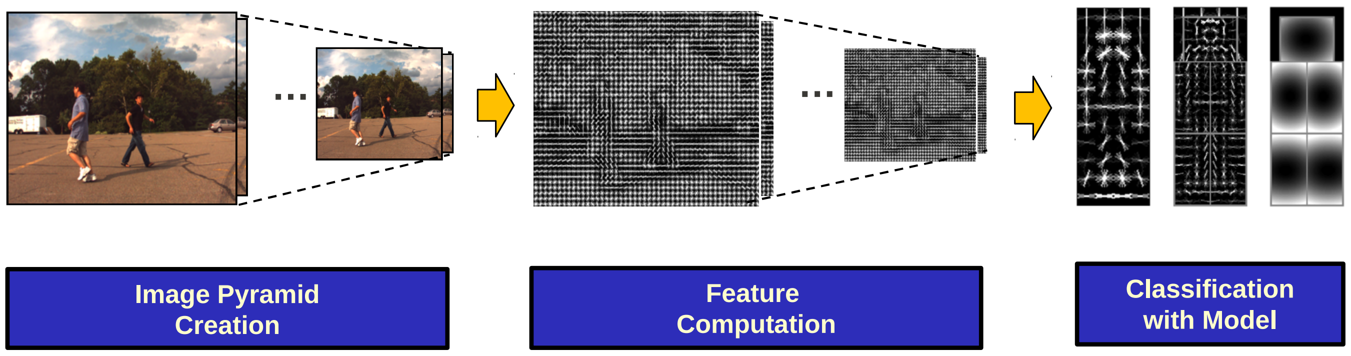


Fig. 2. Illustration of a procedure for a mutiscale pedestrian detection. The main computational bottlenecks are feature computation for densely sampled image pyramid (HOG features in our case) and sliding-window classification (a large number of cross-correlation operations with several parts in our case).

III. Implementation of Deformable Part-Based Models

Our approach is based on the deformable part-based HOG model [11]. We have engineered this algorithm in an attempt to optimize it to be the core of a real-time pedestrian detection for automotive applications. To achieve this, we re-implemented the algorithm (originally based in MATLAB/C) using C and optimized several subsystems in an attempt to improve the algorithms overall speed. The remainder of this section discusses the main reasons for choosing the deformable part-based model and describes the important details of our implementation that were key to its fast performance.

A. *Why Deformable Part-Based Model ?*

Real-time onboard pedestrian detection from a moving vehicle is a challenging task especially when using a monocular camera as the primary (and sole) sensor. Although a number of approaches have been proposed, there are still a number of aspects of this problem that are difficult and can be considered to be as of yet "unsolved." That being said, the accuracy and computational performance of these detectors are improving with each successive publication. We performed a broad survey across the current state-of-the-art pedestrian detectors and ended up with the following methods [22], [11], [5], [1] as candidates for our implementation.

We opted to implement a detector based on the work of Felzenszwalb at al. [11] for the following reasons. First, this method provides an elegant mechanism for handling a wide range of intra-class variability (i.e., various poses of pedestrians) by having multiple submodels (so-called "components"). Furthermore, the deformable part-based model exploits dynamic configurations of its parts which allows for a wider variation in object appearances to be accepted. The importance of this aspect is well illustrated in a recent survey [7]. Secondly, this method has a well-designed learning technique called "latent SVM" which not only can handle a large training set effectively, but also can learn a part-based model without requiring information about exact part labels. Finally, this approach utilizes an efficient detection mechanism called the star-cascade algorithm [10] which makes it suitable for real-time applications.

TABLE I
PROFILING RESULTS FOR ALL IMPLEMENTATIONS (UNIT:MS)

Algorithm Name	**Computer I** Intel Core2 Duo P8800@2.66GHz 2GB RAM		**Computer II** Intel Core i7 2920XM@2.5GHz 16GB RAM	
	Matlab star-cascade	C star-cascade	Matlab star-cascade	C star-cascade
HOG Feature Computation	1145	300	840	80
Sliding Window Classifier	560	320	300	100
Non-Maximal Suppression	24	10	24	5

B. *Implementation Details*

Our implementation follows a standard procedure for processing the image data that consists of first creating a densely sampled image pyramid, computing features at each scale, performing classification at all possible locations, and finally performing non-maximal suppression to generate the final set of bounding boxes. The key steps for this process are illustrated in Figure 2. The key factors that we found for producing the best results for our algorithm include densely sampling the image pyramid, computationally intensive (somewhat tedious) classification in the search space, and the use of discriminative features such as HOG.

Our final goal in terms of speed performance was a speed of 10fps on 640×480 resolution images. To understand where our efforts would need to be applied in the implementation of this algorithm, we profiled the performance of the original MATLAB/C based detectors. These detectors included the `voc-release3` and `voc-release4` with a star-cascade algorithm[1] [13]. The profiling was performed using two evaluation computers that included an Intel Core2

[1]http://www.cs.brown.edu/~pff/latent/, accessed on May 2011

Duo P8800@2.66GHz with 2GB RAM, labeled computer I, and an Intel Core i7 2920XM@2.5GHz with 16GB RAM, labeled computer II. For 640×480 images and 10 scales per octave, `voc-release3` (1 component with 6 parts, multi-threaded version) demonstrated a performance of 0.5 fps and `voc-release4` algorithm with a star-cascade algorithm (1 component with 8 parts, single-threaded version) demonstrated a performance of 0.6 fps on computer I. The details of the profiling result is shown in Table I. As can be seen in this Table, the two main bottlenecks are the computation of HOG features for densely sampled image pyramids and the sliding-window classification. Since the MATLAB/C based detector already uses compiled C code (MEX functions in MATLAB) we needed to approach these bottlenecks by developing parallel (multi-threaded) implementation of the HOG feature computation functions as well as the classification functions. The key details to this implementation are described below:

Feature computation: Given an input image, the first step of the pipeline is to build a densely sampled image pyramid and compute the corresponding feature pyramid. For an image pyramid, we employed the image resize function of the OpenCV 2.0 library. For the HOG feature pyramid, which was the first computational bottleneck, we refactored the algorithm to make use of the `pthread` library. This was possible because the computational process to generate each level of the pyramid is independent of all the others. This solution allowed us to speed the algorithm up by 1 order of magnitude.

Sliding-window classification: For the `voc-release3` algorithm which is based on star-structured models (1 root filter and 6 part filters), this process at a certain pyramid level corresponds to 7 times cross-correlations between each model and that level of feature pyramid and then 6 times generalized distance transforms [12] to combine all part filter responses. In practice, a large number of cross-correlation is the main bottleneck for this step. For this, the original method provides a parallelized correlation scheme with a numerical linear algebra enhanced function. We ported their MEX functions into our implementation. Whereas, `voc-release4` with a star-cascade algorithm provides an efficient way for classification by evaluating parts in a cascaded fashion with a sequence of thresholds. For implementation of this idea, `voc-release4` uses $2(n+1)$ models for a model with $n+1$ parts, where the first $n+1$ models are obtained by sequentially adding parts with simplified appearance models (PCA version of original HOG feature) for faster computation and second $n+1$ models are obtained by sequentially adding parts with its full feature. Our implementation is a little bit different in that we just use $n+1$ cascade models with full HOG feature for ease of implementation and we parallelize the process using *pthread* library. By doing this, we achieve 2X-4X speed improvement.

Non-maximal suppression: After sliding-window classification, we usually get multiple overlapping bounding boxes from detections nearby locations and scales. Non-maximal suppression (NMS) is used for eliminating those repeated detections. Currently, two dominant NMS approaches have been widely used: mean shift mode estimation introduced in [4] and pairwise max suppression introduced in [11]. We implemented the pairwise max suppression scheme due to its simplicity and efficiency.

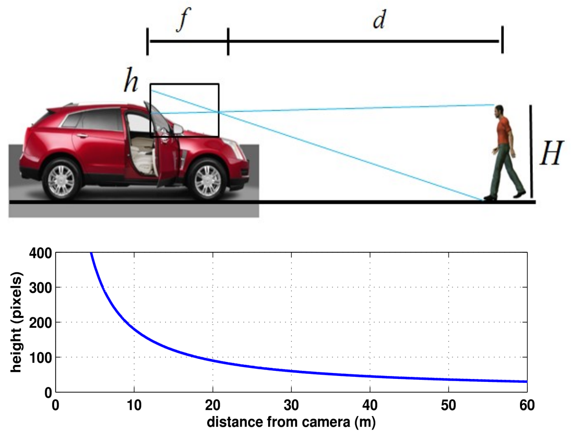


Fig. 3. Geometric constraints analysis. (a) Scene geometry. (b) Pixel height h as a function of distance d.

IV. GEOMETRY ANALYSIS

The use of scene geometry enables two primary benefits for pedestrian detection: efficient and accurate detection. Efficiency can be achieved by searching only geometrically valid regions in the image space and accuracy can be achieved by suppressing a number of potential false positives from the irrelevant image space. Many automotive applications therefore exploit geometric constraints from the known camera calibration information with some assumptions [18], [14]. Here, we propose a simple and efficient search algorithm for our pedestrian detector by analyzing the following relationships: 'pedestrian height in pixels' vs. 'distance from a vehicle' and 'pedestrians' foot position in images' vs. 'distance from a vehicle'. For analysis for these relationships, we assume that 1) ground plane is flat, 2) pedestrians rest on the ground plane, and 3) localization of bounding boxes is accurate (especially for bottom lines of bounding boxes).

Pedestrian height in images: Since Caltech dataset was colleced with a typical automotive camera settings (640×480 resolution, 27^o vertical field of view, and fixed focal length (f) at 7.5mm), we can compute a pixel height h of a pedestrian in a image using a perspective projection model. From Figure 3(a), we can easily draw the relationship between a pixel height (h) and distance (d) as $h \approx H f_p / d$, where H is the true pedestrian height and f_p is the focal length expressed in pixels. This relationship for Caltech dataset is shown in Figure 3(b) assuming $H = 1.8m$.

Depth computation: In general, esimating depth information from monocular camera images is impossible. However, if all assumptions we mentioned above are true, we can compute a range from the camera to a pedestrian. Here, we avoid the derivation of this relationship due to space limitation. The derivation is provided in our project website[2].

We can search an input image efficiently by computing depth information first according to a current $y-coordinate$ of searching step and then selecting geometrically plausible scales of a feature pyramid so that a fixed size pedestrian model can detect real pedestrians. By this search scheme, we can achieve 2X-3X speed improvement.

V. Experimental Results

To quantitatively evaluate our pedestrian detection system, we analyzed its performance on various real-world datasets. To evaluate detection performance, we used the Caltech Pedestrian Dataset [7] which offers an opportunity to exploit many different aspects of model training thanks to its large number of positive samples. The (annotated) dataset corresponds to approximately 2.5 hours of video captured at 30fps. The dataset is segmented into 11 sessions, 6 of which were used for training (S0∼S5) and the rest sessions were used for testing (S6∼S10). We performed a set of experiments to identify the key design parameters for the deformable part-based model. Using the model trained with the optimal parameters from the first set of experiments, we compare our performance with other state-of-the-art detectors using the Caltech Benchmark. This evaluation framework required us to upscale the input images by a factor of 2 (with a final resolution of 1280×960) for the first and second experiments. For subsequent experiments, we fabricated our own set of experiments (what we call `'automotive'`) to better represent data that would be seen from real-time automotive applications.

A. Sytem Design Parameters

Number of Training Samples: The Caltech dataset contains approximately $2,300$ unique pedestrians out of approximately $347,000$ total instances of pedestrians. Because of this high level of redudancy and correlation in the sample images, we had to first determine the best sampling scheme to obtain a statistically valid set of training images. We trained 7 models with a standard set of parameters (1 component 6 parts) where each model trained with a training set (S0∼S5) consisting of images selected from the full data set at a different sampling frequency. We trained models for each of the following frequencies: 1, 10, 20, 30, 40, 50, and 60. We ran each model on the test set (S6∼S10) and evaluated them using the same evaluation code provided from [7] (ver. 3.0.0). The results of these experiments are shown in Table II, where we use log-average miss rate (LAMR) to represent detector performance by a single reference value. The LAMR is computed by averaging miss rate at nine FPPI rates evenly spaced in log-space in the range 10^{-2} to 10^{0} according to [7].

Although there is no large difference in performance between the settings, using every 30th training images (i.e., one training image per one second) in this case gives us the best performance. We decided to use this setting throughout

[2]www.ece.cmu.edu/∼hyunggic/pedestrian.html

TABLE II
Different sampling schemes for model training

Sampling Scheme	No. of Pos. Samples	LAMR (%)
All frames	65570	56
Every 10th	6557	56
Every 20th	3261	55
Every 30th	2198	54
Every 40th	1629	55
Every 50th	1280	56
Every 60th	1088	58

the remainder of our experiments. This result has implication in the generation of ground truth data as annotating every 30^{th} frame of a dataset is far less expensive than having to annotate every frame.

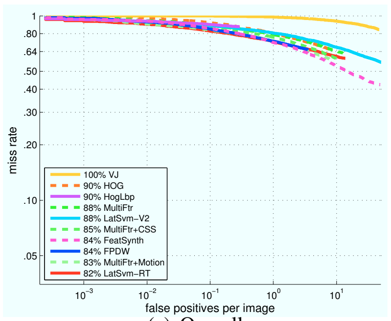
Number of Parts: The standard number of parts for the deformable part-based model is 6 and 8 for `voc-release3` and `voc-release4`, respectively. However, the optimal number of parts depends on the variability of an object class and may be significantly different between classes. Our second experiment was designed to identify the optimal number of parts required for the pedestrian models that would be used for the automotive application. Once again, we trained 7 models with different number of parts (2∼8) using the exact same training set and tested them on the testing set. As shown in Table III, using the standard part number of 6 shows the best performance.

TABLE III
Different number of parts

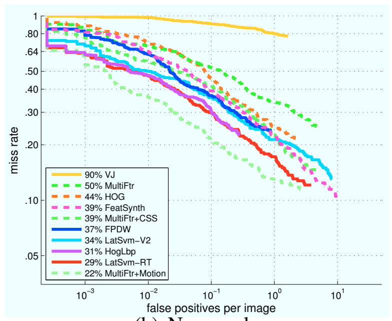
No. of Parts	LAMR (%)
2	58
3	55
4	55
5	56
6	54
7	56
8	56

In general, when selecting the number of parts, say for `voc-release3`, it might be better to use 3 or 4 parts for a faster detection rate. However, for the star-cascade algorithm using 6 to 8 parts seems reasonable due to the cascaded structure of its classifier. To balance efficiency and accuracy, we decided to use 6 parts for our evaluations.

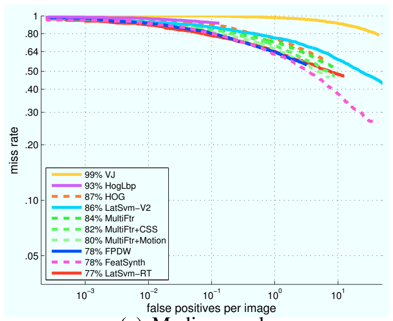
Number of Scales Per Octave: Typically, modern pedestrian detectors use two or three octaves and sample 8-14 scales per octave for accurate detection. Since the computational requirements for feature computation in this image pyramid can be significant, new methods of handling this issue must be identified. Recently, Dollár et al. [5] proposed a technique to avoid constructing such a feature pyramid by approximating feature responses at certain scales by computing feature responses at a single scale. They demonstrated that the approximation is accurate within an entire scale octave. Here, we are interested in primarily looking at performance



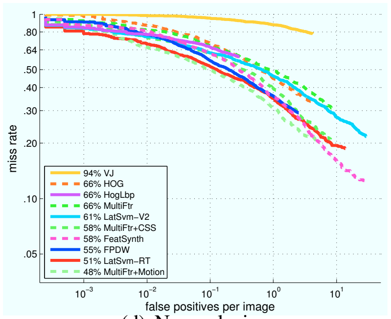
(a) Overall



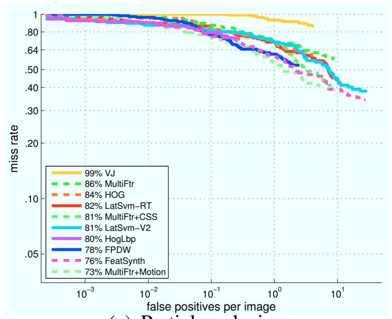
(b) Near scale



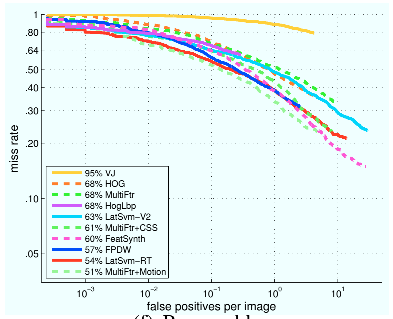
(c) Medium scale



(d) No occlusion



(e) Partial occlusion



(f) Reasonable

Fig. 4. Evaluation results using the same criterions in [7]. (a) Overall performance on all annotated pedestrian images. (b) Performance on unoccluded pedestrians over 80 pixels in height. (c) Performance on unoccluded pedestrians between 30-80 pixels in height. (d) Peformance on unoccluded pedestrians over 50 pixels in height. (e) Same as (d) but with partial occlusion. (f) Performance on pedestrians at least 50 pixels in height under no or partial occlusion.

differences depending on different number of scales and looking for a specific optimal solution for our configuration. We tested 7 different settings and the results are shown in Table IV. We found that even 4 scales per octave shows a marked improvement over 2 scales per octave in accuracy. We decided to use 4 scales per octave as a trade-off between accuracy and performance.

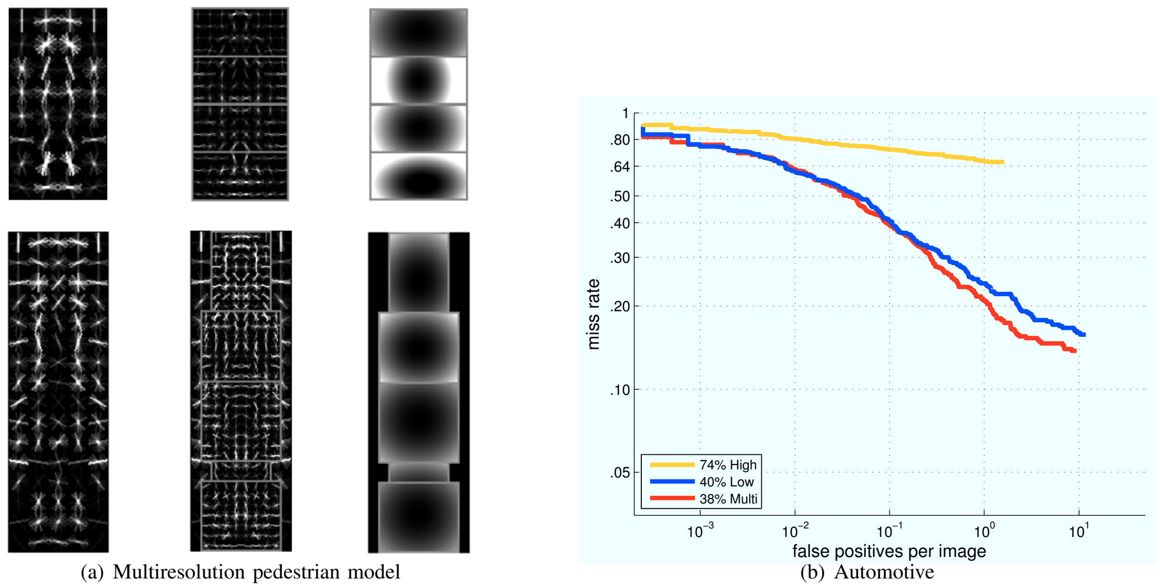
TABLE IV
DIFFERENT NUMBER OF SCALES PER OCTAVE

Scales / Octave	No. of Levels	LAMR (%)
2	10	61
4	19	56
6	28	56
8	37	55
10	46	54
12	56	54
14	65	53

B. *Evaluation with Caltech Benchmark*

Using a model parameterized with the values identified in our previous experiments, we compared the performance of our detector with those of other state-of-the-art detectors using the Caltech evaluation framework. This framework provides a flexible way to evaluate detectors under various conditions on mutiple datasets. We evaluated the performance of the algorithms under six conditions on the testing data in the Caltech dataset. Although the framework provides detection results from sixteen pre-trained detectors to help provide consistent comparison, we intentionally selected 10 of the most relevant top-performing detectors to increase the clarity of the resulting graphs. The criterion we used to select the algorithms was detection accuracy and runtime. One exception is the "MultiFtr+Motion" algorithm, which shows very slow detection speed (estimated around 30 seconds for a 640×480 image on our evaluation computer II). But we included the method since it is the only method which uses motion features in our list and because it shows the best performance in most cases. We named our algorithm "LatSvm-RT". Following the naming convention for the Caltech benchmark we described the other detectors as follows: VJ [19], HOG [4], HogLbp [22], MultiFtr [23], LatSvm-V2 [11], MultiFtr+CSS [21], FeatSynth [1], FPDW [5], MultiFtr+Motion [21]. All detectors except MultiFtr+CSS [21] and MultiFtr+Motion [21] (which use their own dataset called 'TUD-MP') were trained with the "INRIA Person Dataset [4]." Note that the LatSvm-V2 is our baseline detector, `voc-release3`. Our results are shown in Figure 4. Following the example found in [7], we plot miss rate vs. false positives per image (FPPI) and use the log-average miss rate as a single comparison value for the overall performance. The entries are ordered in the legend from the worst log-average miss rate to the best.

In general, our detector shows very good performance except for the "partical occlusion" test case. Figure 4(a) illustrates algorithmic performance on the entire test set which contains pedestrians from 20 pixels in height which are in general fairly poor. Results for near and medium scale unoccluded pedestrians, corresponding to heights of at least



(a) Multiresolution pedestrian model (b) Automotive

Fig. 5. Evaluation results using the *automotive* criterion. (a) Multiresolution pedestrian model: high-resolution (128×40) with 475 positive samples and row-resolution (64×32) with 2430 positive samples. (b) Performance on unoccluded pedestirans over 70 pixels tall.

80 pixels and 30-80 pixels, respectively, are shown in Figure 4(b) and Figure 4(c). In each case, our detector shows the second best or best performance with a log-average miss rate of 29% and 77%, respectively. As can be seen in Figure 4(d) and Figure 4(e), performance drops significantly under partial occlusion, leading to a dramatic decrease of a log-average miss rate from 51% to 82% for our case. This is quite disappointing since our classification algorithm can not handle partial occlusions even though the part-based model itself has a rich representation for partial occlusion. We will seek to address this through the development of a new classification algorithm for partial occlusion handling in our future work. Finally, Figure 4(f) shows performance on pedestrians over 50 pixels tall under no or partial occlusion. Here our detector shows second best performance with a log-average miss rate of 54%.

C. Real-Time Evaluation

Because the goal of our research is to develop pedestrian detection algorithms that are suitable for the rigors of deployment on an autonomous vehicle, we developed a new test scenario that we call *'automotive'* to show the performance of our detector in this context. We define real-time operation to be 10fps@640×480. For the automotive experiment, the *'reasonable'* scenario in the Caltech benchmark does not fit to our needs for several reasons. First, it uses detection results from upscaled input images to compare performances. However, it is almost impossible to process these images at 10Hz without an unreasonable amount of computational power. Second, we need to adjust ground truth information according to our working distance. In the *'automotive'* setting, we can only include unoccluded pedestrians within 25m from a vehicle (corresponds to 70 pixels in height) in the ground truth. To increase the working range of our detector while maintaining a high detection rate, we trained a multiresolution pedestrian model (shown in Figure 5(a)). We intentionally set the height of our low-resolution model as 64 pixels so that our system can detect pedestrians up to 25m reliably. We run our detector on the Caltech testset without upscaling (i.e., 640×480 images) with the efficient search scheme discussed in Section IV. We achieve a real-time operation at 14fps with a log-average miss rate of 38%. We believe our work is a meaningful progress of state-of-the-art pedestrian detectors. The performance is shown in Figure 5(b) and some qualitative results achieved on the Caltech testset are shown in Figure 6.

VI. CONCLUSIONS AND FUTURE WORK

This paper presents a real-time pedestrian detection system for intelligent vehicles using a deformable part-based model. For support real-time operation, we implement two different versions of Felzenszwalb et al.'s object detection systems (a baseline [11] and a star-cascade method [10]). Scene geometry from known camera calibration information is utilized to search a feature pyramid more efficiently. For better detection accuracy, a multiresolution pedestrian model is used for detecting small (pixel-sized) pedestrians as well as normally-sized ones. Using the Caltech Pedestrian Dataset, we quantitatively evaluated our detection system with a series of experiments and showed real-time operation (14fps@640x480 images) while maintaining the state-of-the-art detection accuracy (80% detection rate with 1 FPPI) under our test scenario called *'automotive'*. As part of our future work we want to develop a partial occlusion handling algorithm to increase detection accuracy.

VII. ACKNOWLEDGMENTS

This project was funded by General Motors through the General Motors-Carnegie Mellon Autonomous Driving Col-

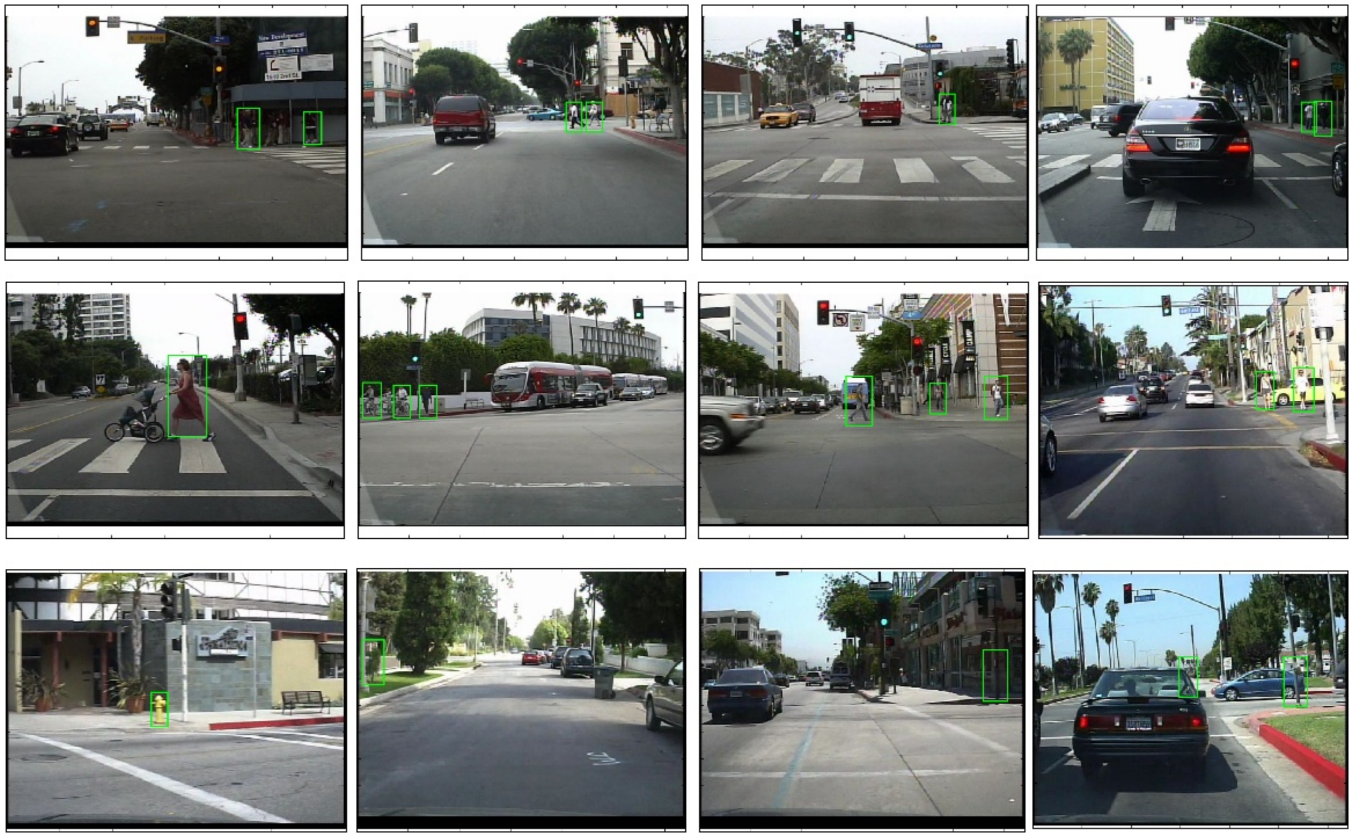


Fig. 6. Qualitative detection results on the Caltech testset. The first and second row shows correct pedestrian detections in various scenarios. The third row shows typical false positives.

laborative Research Lab.

REFERENCES

[1] A. Bar-Hillel, D. Levi, E. Krupka, and C. Goldberg. Part-based feature synthesis for human detection. In *ECCV*, 2010.

[2] H. Cho, P. Rybski, and W. Zhang. Vision-based bicycle detection and tracking using a deformable part model and an ekf algorithm. In *ITSC*, 2010.

[3] H. Cho, P. Rybski, and W. Zhang. Vision-based 3d bicycle tracking using deformable part model and interacting multiple model filter. In *ICRA*, 2011.

[4] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005.

[5] P. Dollár, S. Belongie, and P. Perona. The fastest pedestrian detector in the west. In *BMVC*, 2010.

[6] P. Dollár, Z. Tu, P. Perona, and S. Belongie. Integral channel features. In *BMVC*, 2009.

[7] P. Dollár, C. Wojek, B. Schiele, and P. Perona. Pedestrian detection: An evaluation of the state of the art. *PAMI*, 99(PrePrints), 2011.

[8] M. Enzweiler and D. Gavrila. Monocular pedestrian detection: Survey and experiments. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 31:2179–2195, 2008.

[9] C. U. et al. Autonomous driving in urban environments: Boss and the urban challenge. *Journal of Field Robotics, Special Issue on the 2007 DARPA Urban Challenge, Part I*, 25(8):425–466, 2008.

[10] P. Felzenszwalb, R. Girshick, and D. McAllester. Cascade object detection with deformable part models. In *CVPR*, 2010.

[11] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 99, 2010.

[12] P. Felzenszwalb and D. Huttenlocher. Distance transforms of sampled functions. *Cornell Computing and Information Science Technical Report TR2004-1963*, 2004.

[13] P. F. Felzenszwalb, R. B. Girshick, and D. McAllester. Discriminatively trained deformable part models, release 4. http://people.cs.uchicago.edu/∼pff/latent-release4/.

[14] D. M. Gavrila and S. Munder. Multi-cue pedestrian detection and tracking from a moving vehicle. *International Journal of Computer Vision*, 73:41–59, 2007.

[15] D. Gerońimo, A. Lopéz, and T. G. A. Sappa. Survey of pedestrian detection for advanced driver assistance systems. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 32, 2010.

[16] K. Mikolajczyk, C. Schmid, and A. Zisserman. Human detection based on a probabilistic assembly of robust part detectors. In *ECCV*, 2004.

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

[18] A. Shashua, Y. Gdalyahu, and G. Hayun. Pedestrian detection for driving assistance systems: Single-frame classification and system level performance. In *IEEE IV*, pages 13–18, 2004.

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

[20] P. Viola, M. J. Jones, and D. Snow. Detecting pedestrians using patterns of motion and appearance. *International Journal of Computer Vision*, 63(2):153–161, 2005.

[21] S. Walk, N. Majer, K. Schindler, and B. Schiele. New features and insights for pedestrian detection. In *CVPR*, 2010.

[22] X. Wang, T. X. Han, and S. Yan. An hog-lbp human detector with partial occlusion handling. In *ICCV*, 2009.

[23] C. Wojek and B. Schiele. A performance evaluation of single and multi-feature people detection. In *DAGM Symposium Pattern Recognition*, 2008.

[24] C. Wojek, S. Walk, and B. Schiele. Multi-cue onboard pedestrian detection. In *CVPR*, 2009.