Detecting Pedestrians at Very Small Scales

Luciano Spinello, Albert Macho, Rudolph Triebel and Roland Siegwart

Autonomous Systems Lab, ETH Zurich, Switzerland
email: {luciano.spinello, rudolph.triebel, roland.siegwart}@mavt.ethz.ch

***Abstract*— This paper presents a novel image based detection method for pedestrians at very small scales (between 16 x 20 and 32 x 40). We propose a set of new distinctive image features based on collections of local image gradients grouped by a *superpixel* segmentation. Features are collected and classified using AdaBoost. The positive classified features then vote for potential hypotheses that are collected using a mean shift mode estimation approach. The presented method overcomes the common limitations of a sliding window approach as well as those of standard voting approaches based on interest points. Extensive tests have been produced on a dataset with more than 20000 images showing the potential of this approach.**

I. Introduction

From the different participants in typical urban traffic scenarios, pedestrians are the most vulnerable ones as they are not protected by any kind of equipment as they exist for motorists and cyclists. This fact is lamentably reflected in the annual traffic accident statistics, as they are published, e.g. by the Touring Club Switzerland (TCS) [1]. Here, two major trends can be observed: first the steady decrease in the total number of dead and seriously injured persons over the last 30 years, and second the increase in the percentage of dead and injured pedestrians. The former is mostly due to the growing number of safety systems available for modern vehicles, while the latter originates from the fact that primarily motorists and cyclists benefit from such safety systems, but not pedestrians. One way to address this problem is to build more intelligent driver assistant systems that aim at protecting the driver *and* the pedestrian and avoid a potential collision. A major requirement for this is, of course, the reliable detection of pedestrians in urban traffic environments. However, this task is rendered particularly difficult by at least the following two facts:

- Pedestrians show a very high variability in shape and color due to physical size, clothing, carried items, etc.
- In urban environments, especially in city centers, pedestrians most often appear in large numbers, e.g. when crossing at a traffic light. This results in many occlusions where the pedestrians are only partly visible.

Despite these difficulties, there are already some encouraging approaches to detect pedestrians, majorly based on camera data (e.g.[15]), but also using 2D laser range scanners [2] or both [23]. However, these systems require a certain minimal size at which the pedestrians are visible in the data, which has the drawback that pedestrians that are far away, as well as children, can not be detected reliably. According to the rule of thumb from theoretical traffic lessons, a car that moves with $50km$ per hour needs $40m$ to come to a full stop. This is still far from the maximal distance at which pedestrians can be detected with current approaches, using a lens that provides still an acceptable opening angle (above 90 degrees). In this paper, we present an approach to detect pedestrians that are up to $50m$ away while the lens still provides a wide field of view. The size in which the pedestrians appear in the image is as low as 16 by 20 pixels. Our proposed technique uses a supervised learning algorithm consisting of the following two major steps:

Fig. 1. Detection of very small scale pedestrians in a urban walkway.

- **Training** Based on a *superpixel* segmentation proposed by Felzenszwalb and Huttenlocher [9] and a computation of the image gradient, segments of strong edge pixels are extracted. From these edge segments $\mathbf{s}$, we extract feature vectors based on a combination of histograms of gradient orientations and the angles that each line segment from a polyline approximation of $\mathbf{s}$ forms with the horizontal axis. These features are used to train an AdaBoost classifier [11]. In addition, we store the positive training examples in a *codebook* together with their displacement vectors with respect to the object centers. This is inspired by the voting scheme of the Implicit Shape Model (ISM) approach (see [15]).
- **Classification** Again, we compute edge segments and feature vectors. Then we run the classifier and collect all votes for object centers that are cast from edge segments

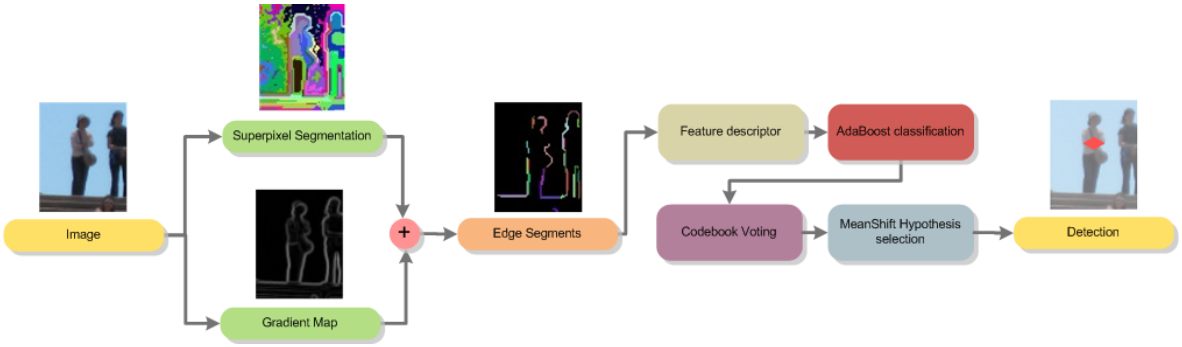

Fig. 2. Flowchart of our detection algorithm.

classified as positive. Using mean shift mode estimation [4], we obtain object positions for which many edge segments vote and thus are strong hypotheses for the position of an object, i.e. a pedestrian (see also Fig. 2).

Our approach avoids both the necessity of a sliding window, as e.g. in [24], [5], and the requirement of a minimal number of detected interest points (e.g. Hessian or Harris corners [17]) to obtain robust hypotheses for small objects such as in [15], [22]. We present the following novelties:

- the segmentation of edges from the gradient image using a superpixel segmentation. This divides the edges into chunks of homogeneous gradient variability and provides highly informative local edge features. This overcomes the usage of an overlapping tessellation to learn object features and uses a more semantical subdivision: at these image sizes, superpixels tend to segment persons into more meaningful parts like torso, head, limbs. The reason for that is that, due to the smaller resolution, the gradient variability is usually lower than at higher scales.
- a novel image descriptor particularly suited for the detection of objects at small scales,
- a classifier based on a combination of AdaBoost and the voting scheme known from the ISM approach.

The paper is organized as follows. In the next section we discuss approaches from the literature that are most closely related to our method. In Sec. III, we describe the feature extraction and the details of our edge descriptor. Then, in Sec. IV we present our classification technique and the hypothesis generation. Sec. V shows the experimental results and in Sec. VI we draw our conclusions.

II. Related Work

In the area of image-based people detection, there mainly exist two kinds of approaches (see [18] for a survey). One uses the analysis of a *detection window* [5] or *templates* [12], [24], the other performs a *parts-based* detection [8], [13]. Leibe *et al.* [15] presented an image-based people detector using *Implicit Shape Models* (ISM) with excellent detection results in crowded scenes. An extension of this method that proposes a feature selection enhancement and a nearest neighbor search optimization has been already shown in [22][23]. In the specific area of small scales pedestrian detection very few works are present. Viola *et al.* [24] detect small pedestrians (bigger than the ones detected in this paper) including a time integration. Efros *et al.* [6] uses optical flow reasoning to detect humans and understand actions. Ferrari *et al.* [10] classify contours for detecting simple objects (coffee mugs, animals) in clutter by using an iterative path search among linked segments. The superpixel method has been introduced by Ren and Malik [20] using a Normalized Cut criterion [21] to recursively partition an image using contour and texture cues. Other methods have been proposed to obtain quality superpixel segmentations [9], [7].

III. Feature Extraction

In the literature, many different approaches are presented to compute local interest point detectors and appropriate region descriptors (for a comparison see [17]). However, for our particular problem of object detection at very small scales, none of these approaches is well suited for the following reasons:

1) In areas of many small objects, usually – if at all – only few interest points such as Harris corners or Hessian blobs can be detected. Thus, the number of possible voters is very low compared to the number of objects to be detected. This results in detection results with low confidence. We therefore decided to use edges instead of interest points, as described below.
2) Standard descriptors such as SIFT [16], Histogram of Oriented Gradients (HOG) [5], and shape context [3] represent the local information in a high dimensional feature space. One could think of applying such descriptors to all (or some) points of an edge chain, but this would result in a large number of feature dimensions. Given that the size of the objects to be detected usually ranges only about 300 pixels, this seems inappropriate.

As a conclusion, we aim at finding a simple but informative descriptor that is defined on chains of edge pixels and can be computed efficiently. The decision to use chains of edge pixels or, as we will denote them, *edge segments*, is somehow inspired by the use of *edgelets* for detecting pedestrians (see [25]). In the following, we present the details of our method to compute edge segments and local descriptors.

A. Superpixel Segmentation

The aim of this first step of our detection algorithm is to preprocess a given input image and to obtain a more semantic representation that is independent on the pixel resolution of the image. One common way to achieve that is by grouping image pixels into regions in such a way that all pixels in a region are similar with respect to some kind of similarity measure. In the literature, this is also known as *image segmentation*, and it is crucial for a large number of algorithms and applications in computer vision. Many different algorithms have been suggested for this problem and we refer to the related work section in [9] for a good overview. Two of the more recent and mostly used approaches, namely [20] and [9], define a graph where the nodes are the image pixels and the graph edges are defined by a neighbor relationship between pixels. Of these two, the approach by Felzenszwalb and Huttenlocher [9] is more tuned for computational efficiency and the one by Ren and Malik [20] is more robust and yields more informative regions. For our application of detecting small scale objects, the use of complex similarity measures such as the peaks in contour orientation energy as in [20] is not required. Therefore, we use the former approach in our framework. This algorithm groups the pixels into segments so that the minimal dissimilarity *across* two segments is still higher than the maximal dissimilarity *within* both segments. The number of produced segments – usually named *superpixels* – can be adjusted by a parameter k. An important characteristic of this method is its ability to preserve details in low-variability image regions while ignoring details in high-variability regions. Therefore, it is especially suited for our application, because pedestrians at small scales are usually represented by only very few pixels in which the color variability is comparably low due to the lower sampling resolution. This means that one superpixel often represents an entire body part like a leg, a torso, or a head.

B. Edge Segments and the Edge Descriptor

As mentioned above, we need to find a descriptor that is not only assigned to single interest points, as those occur less frequently in areas of small scale objects. Using superpixels as regions of interest are a much better choice here, as they are always found and they represent a higher vicinity. However, defining a region descriptor for superpixels would result in very complex computations. For our purpose, this is not appropriate, as we only want to represent the information contained in small image regions. As a tradeoff between single pixels and regions, we use *edge segments*, which are defined as chains of edge pixels that lie inside a superpixel. For the computation of the edge pixels, we apply the Sobel operator to the grayscale image and remove edges with a gradient magnitude that is below a threshold τ. From that, we compute the edge segments by simply applying the superpixel segmentation described above to the edge image.

Adapted to our choice of edge segments we define a descriptor that reflects the local information of each edge segment. This information is later used for our object detection algorithm. In accordance to the notion of a region descriptor, we refer to this as an *edge descriptor*. In our experiments, we tested the following two kinds of edge descriptors:

- **Histogram of orientations:** The local gradient orientations along an edge segment are collected in a histogram with n bins: each bin B_i counts the number e_i of edge points $\mathbf{p}$ at which the gradient $\gamma(\mathbf{p})$ has a certain orientation (see Fig. 4, left). For the descriptor, we use $2n$ values, where the first n are the values e_i, normalized by the sum $m := \sum_{i=1}^{n} e_i$, and the second n values are the sums $\sum_{\mathbf{p}_j \in B_i} |\gamma(\mathbf{p}_j)|$ for each bin B_i, again normalized by m. We name this descriptor HIST.
- **Vector of directions:** First we compute for each edge segment a polyline approximation consisting of l line segments. We do this using a variant of split-and-merge. Then, we collect all angles between the line segments and the horizontal axis in a vector of length l (see Fig. 4, right). We name this descriptor VECT.

IV. Feature Classification

Based on the feature extraction described in the previous section, our goal is to formulate an algorithm that classifies these feature vectors into one of the two classes 'pedestrian' or 'background'. For this task, we employ a supervised learning technique that uses a hand-labeled training data set with positive examples of small scale pedestrians. Many techniques have been proposed to achieve this task. Two very successful approaches are the face detection algorithm of Viola and Jones [24] and the voting technique named Implicit Shape Model (ISM) by Leibe *et al.* [15]. The advantage of the first method is the strength of AdaBoost [11], i.e. a classifier that is arbitrarily accurate on the training data and at the same time yields a rating of the most relevant feature dimensions for classification. The downside is that the image has to be searched with a sliding window approach and at different scales. In contrast, the voting scheme of ISM relies on scale invariant features that are stored in a codebook along with the relative position of the object center. No feature search is needed here in the image, but the algorithm does

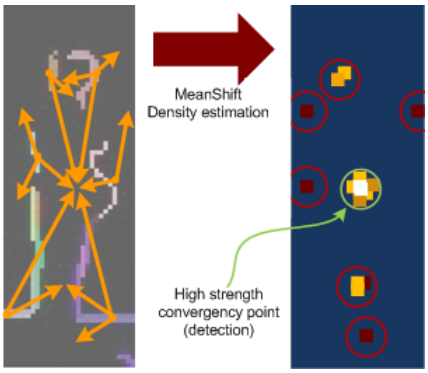

Fig. 3. Visual explanation of codebook voting. 1) Matched descriptors vote for different center positions 2) Mean Shift mode estimator is run in order to converge in local high density areas in the voting space 3) High strength hypotheses are selected as detections.

not rank certain feature dimensions over others when finding matches in the codebook. Thus, extracted feature vectors may vote for a potential object center, even though they reveal a low evidence for the occurrence of the object.

In this paper, we suggest to combine both ideas to a method that pre-classifies a given feature vector using AdaBoost and then, in the case a positive classification, searches for a vote of the object center in the codebook. The details of this are described in the following.

A. AdaBoost Classification

Boosting is a method to combine a set of weak classifiers into a strong classifier. The only requirement for a weak binary classifier is that its classification error on any given training data set is bigger than 0.5, i.e. it must be better than random guessing. Strong classifiers, however, can reach arbitrary low training error rates. AdaBoost [11] achieves this by adaptively assigning weights to the training examples and iteratively learning M weak classifiers h_i and corresponding weights α_i. After learning, the sum

$$g(\mathbf{z}) := \sum_{i=1}^{M} \alpha h_i(\mathbf{z}) \tag{1}$$

is used to decide whether a given test feature $\mathbf{z}$ is classified as positive or negative by simply taking the sign of the result of g. A broadly used type of weak classifiers are *decision stumps*, and we also use them in our framework. A decision stump finds a hyperplane η in feature space that is perpendicular to one feature dimension. It is uniquely defined by the index of the feature dimension, the orientation of the normal vector of η, and the distance of η to the origin.

The features extracted in the previous step are expressed as a $2n$ dimensional point for the first case and as a l dimensional point in the second case. Features quality are evaluated by learning a classifier for each kind of descriptor. Moreover, we measured the quality of the combination of descriptor VECT with HIST concatenating their values in a single feature of dimension $2n + l$. We call this descriptor MIX.

B. Descriptor Codebook

The main idea of voting based classification techniques, such as the one described by Leibe *et al.* [15], is to collect a set of image descriptors together with displacement vectors, usually named *votes*, and to store them into a *codebook*. The justification of this is that each descriptor can be found at different positions inside an object. Thus, a vote points from the position of the descriptor to the center of the object as it was found in the training data. To obtain a codebook from labeled training data, all descriptors are clustered, usually using agglomerative clustering, and the cluster centers are stored, along with all votes corresponding to a particular cluster. For the detection, new descriptors are computed on a test image and matched against the descriptors in the codebook. The votes that are cast by each matched descriptor are collected in a *voting space*, and a mean-shift maximum density estimator is used to find the most likely position of an object (see Fig. 3).

Fig. 4. The two types of edge descriptors used in our detection algorithm. **Left:** Histogram of orientations: for each edge point of a segment we use the orientations of the corresponding gradient, here shown in yellow on four sample edge points, and compute a histogram over them. **Right:** Vector of line segment orientations: From a polyline approximation to the edge segment, here shown as yellow arrows, we store the orientation angles of each line segment in a vector.

C. Detecting Pedestrians

Once the AdaBoost classifier is trained and a codebook is created from the training data, our detection algorithm proceeds as follows. For a given input image, the gradient map and the superpixel segmentation is computed. Using the latter ones, we obtain the edge segments of the test image. Then, we compute the descriptors as described above and apply AdaBoost using equation (1). All descriptors that are classified positive, are matched to the entries in the codebook. Here, we do a range search to find all descriptors $\mathbf{d}$ that are within a given Euclidean distance r from the query descriptor $\mathbf{d_q}$. Then, all the votes cast from these descriptors are collected in the voting space by adding their displacements to the centroid of the edge segment for which $\mathbf{d_q}$ was computed. In the last step, we apply mean shift mode estimation [4] in the voting space to find the most likely object center for the given votes. Here, we set the kernel radius to half of the width of the training images. To initialize the mean shift estimation, we first collect all votes in a 2D histogram with $0.5w \times 0.5h$ bins where w and h are the width and height of the test image, and then start mean shift at the position of the biggest bins. After convergence of mean shift, we obtain all object hypotheses. From these, we retain those that have a minimum number of votes τ_v.

V. Experiments

To evaluate our detection algorithm quantitatively, we applied it on a large set of test images with labeled positive and negative examples. We trained our classifier with images from pedestrians in two sizes, namely 16×20 and 32×40 pixels. This corresponds in our case to an approximate distance of $56m$ and $28m$, respectively (the focal length of our lens is $4.2mm$).

A. Setting the Parameters

As mentioned before, our algorithm depends on several parameters: the superpixel coarseness k, the gradient strength threshold τ, the length of the descriptor vectors m and n, and the distance parameter r for codebook clustering. To determine these parameters, we created a validation dataset of 2000 random images and evaluated 25 combinations of these parameters on these images. To limit the parameter

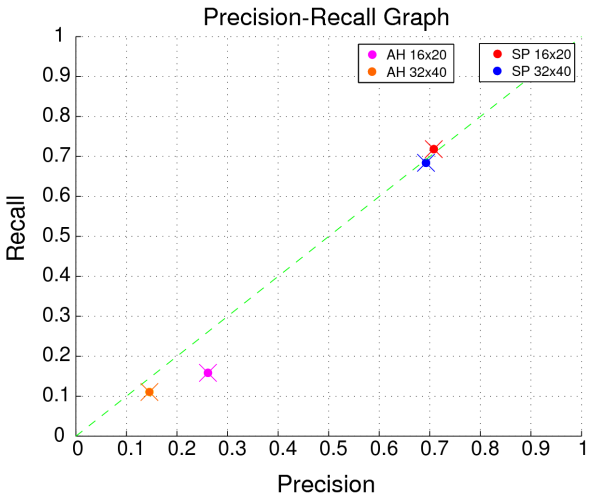

Fig. 5. Precision-Recall graph for our detection algorithm (SP) and for AdaBoost with Haar features as in [14] (AH). Due to the special design of our descriptor for small scales, the detection on the smaller images yields better results. The green line depicts the Equal Error Rate (EER)

search space, we chose m and n from the interval $[6, 8]$ as described in [5], and τ from $[0, 40]$ to ensure that at most 15% of the gradient information is lost (considering that the maximal possible value is 255). The parameter combination with the maximal sum of true positive and true negative detections was used in the later experiments. We obtained $k = 25$, $m = 8$, $n = 8$, $r = 18$ and $\tau = 25$.

B. Training

For training, we used an internationally standard dataset: the NICTA large pedestrian image dataset [19]. It contains pictures with pedestrians taken in typical urban environment. They appear either alone or in crowded scenes whith partial occlusions, in different poses (walking, standing, sitting) and in a broad range of lighting variations. Negative examples are represented by random crops of images from indoor and outdoor environments.

We randomly selected 10000 positive images and 50000 negative images for each scale. In total we trained our algorithm with 120000 image samples. In each image we encountered between 10 and 20 edge segments, i.e. several million descriptors were used for training. We used 5 times more negative training examples to provide a large variety of background. To assess the quality of the AdaBoost training we used a *leave-one-out* cross validation, in which data is partitioned into subsets such that the analysis is initially performed on a single subset, while the other subsets are retained for confirming and validating the initial results. The training was performed on a quad core Intel Xeon CPU with 4GB of RAM in several hours of processing time.

C. Quantitative Results

The test set is composed of 24000 images with 4000 and 20000 negative examples. We evaluated our algorithm for three different kinds of edge descriptors (see Sec. III-B): Histogram of orientations (HIST), Vector of directions (VEC), and both (MIX). The evaluations of the three types of classifiers (HIST, VECT, and MIX) are shown in tables I–III for image size 16×20 and in tables IV–VI for 32×40. The precision-recall values are depicted in Fig. 5, along with the result from the full-body detector for 14×28 images by Kruppa *et al.* [14]. This method, outperformed by our technique, uses AdaBoost with Haar features and is very similar to the one that is described by Munder and Gavrila [18] as close to the best.

The VEC descriptor yields a much lower True Positive Rate (TPR) than the HIST descriptor, which is most probably due to the information loss caused by the polyline approximation of an edge segment. Note that the False Positive Rate (FPR) of both descriptors are similar. The best results are obtained using the combination (MIX) of both descriptors that improves each statistics. It is important to remark that the results for images of size 16×20 are generally better than for those of size 32×40. The reason for this the specific design of our feature descriptors: a bigger image scale tends to exhibit a higher level of detail, therefore the superpixel segmentation yields edge segments that are less distinctive compared to those from the low scale images. In the latter ones, superpixels represent body parts at a higher semantic level (legs, heads, arms), whereas at larger scales, the superpixels are less informative. Moreover, due to the fact that low scale images have a lower resolution, mainly strong edge pixels prevail. This means that thresholding the gradient map at the same value τ results in a lower loss of information. Nevertheless, the proposed technique performs comparably well for images at higher scale: the TPR is only about 3% and the TNR is only about 5% lower.

As a qualitative result, we show in Fig 6 the detection result from a fraction of the test data set at image size 16×20. All images are arranged in a grid and the estimated object centers are depicted with yellow dots.

VI. Conclusions

We presented a novel image based detection method for pedestrians at very small scales. For this particular problem with sparse visual information we propose a new feature descriptor inspired by edgelets in combination with superpixel segmentation. Our technique overcomes common drawbacks of the standard interest point voting approach and of the scrolling window approaches using a descriptor codebook and a robust AdaBoost classification technique. We have evaluated parameters and show quantitative results on a large dataset, showing the effectiveness of our method. In future works we want to investigate how an intelligent tracking can improve the results and how to improve the feature robustness with respect to the scale magnification.

VII. Acknowledgments

This work was funded within the EU Projects BACS-FP6-IST-027140 and EUROPA-FP7-231888.

TABLE I

CONFUSION MATRIX FOR SIZE 16X20 - HIST DESCRIPTOR

Ground truth	Prediction P	Prediction N
P	70.5%	29.5%
N	18.0%	82.0%

TABLE II

CONFUSION MATRIX FOR SIZE 16X20 - VECT DESCRIPTOR

Ground truth	Prediction P	Prediction N
P	56.6%	43.4%
N	18.8%	81.2%

TABLE III

CONFUSION MATRIX FOR SIZE 16X20 - MIX DESCRIPTOR

Ground truth	Prediction P	Prediction N
P	71.8%	28.2%
N	17.1%	82.9%

TABLE IV

CONFUSION MATRIX FOR SIZE 32X40 - HIST DESCRIPTOR

Ground truth	Prediction P	Prediction N
P	66.7%	33.3%
N	23.8%	76.2%

TABLE V

CONFUSION MATRIX FOR SIZE 32X40 - VECT DESCRIPTOR

Ground truth	Prediction P	Prediction N
P	53.0%	47.0%
N	23.9%	76.1%

TABLE VI

CONFUSION MATRIX FOR SIZE 32X40 - MIX DESCRIPTOR

Ground truth	Prediction P	Prediction N
P	68.4%	31.6%
N	22.1%	77.9%

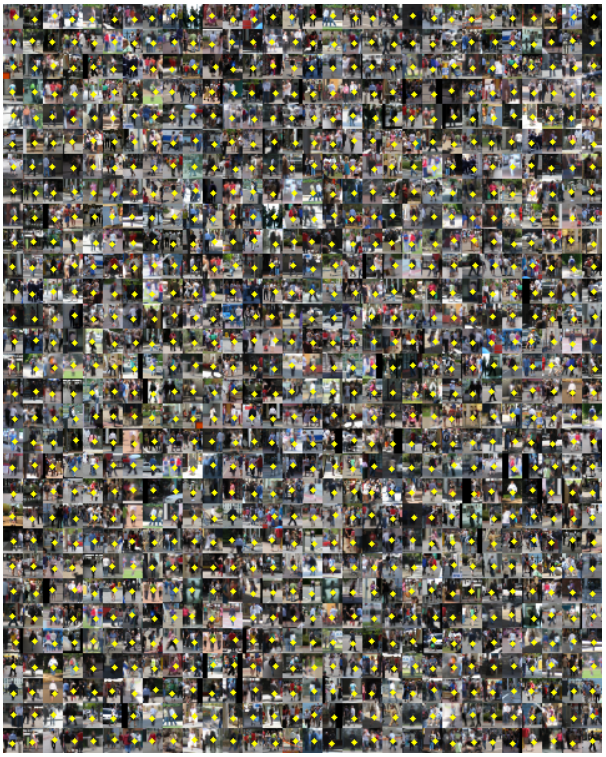

Fig. 6. Qualitative result of our detection algorithm. 600 full size images of correct detections from the test dateset are shown here in matrix form. The yellow dots are the estimated object centers. To keep the presentation uncluttered, the detected bounding box for each image is not displayed.

REFERENCES

[1] http://www.tcs.ch/main/de/home/sicherheit/infrastrukturen/statistik_unfalle.html. (in german).

[2] K. O. Arras, Ó. M. Mozos, and W. Burgard. Using boosted features for the detection of people in 2d range data. In *IEEE Int. Conf. on Rob. & Autom. (ICRA)*, 2007.

[3] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Trans. on Pattern Analysis & Machine Intell.*, 24:509–522, 2002.

[4] D. Comaniciu, P. Meer, and S. Member. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24:603–619, 2002.

[5] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *IEEE Conf. on Comp. Vis. and Pat. Recog. (CVPR)*, pages 886–893, Washington, DC, USA, 2005. IEEE Computer Society.

[6] A. A. Efros, A. C. Berg, G. Mori, and J. Malik. Recognizing action at a distance. In *ICCV*, pages 726–733, Nice, France, 2003.

[7] D. Engel, L. Spinello, R. Triebel, R. Siegwart, H. Bülthoff, and C. Curio. Medial features for superpixel segmentation. In *Proc. of Machine Vision and Applications*, 2009.

[8] P. Felzenszwalb and D. Huttenlocher. Efficient matching of pictorial structures. In *IEEE Conf. on Comp. Vis. and Pat. Recog. (CVPR)*, pages 66–73, 2000.

[9] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient graph-based image segmentation. *Int. Journ. of Comp. Vis.*, 59(2):167–181, 2004.

[10] V. Ferrari, T. Tuytelaars, and L. Van Gool. Object detection by contour segment networks. In *European Conf. on Computer Vision (ECCV)*, pages III: 14–28, 2006.

[11] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.

[12] D. Gavrila and V. Philomin. Real-time object detection for "smart" vehicles. In *IEEE Int. Conf. on Computer Vision (ICCV)*, 1999.

[13] S. Ioffe and D. A. Forsyth. Probabilistic methods for finding people. *Int. Journ. of Comp. Vis.*, 43(1):45–68, 2001.

[14] H. Kruppa, M. Castrillon-Santana, and B. Schiele. Fast and robust face finding via local context. In *Joint IEEE Intern. Workshop on Vis. Surv. and Perform. Eval. of Tracking and Surv.*, 2003.

[15] B. Leibe, E. Seemann, and B. Schiele. Pedestrian detection in crowded scenes. In *IEEE Conf. on Comp. Vis. and Pat. Recog. (CVPR)*, pages 878–885, Washington, DC, USA, 2005. IEEE Computer Society.

[16] D. Lowe. Distinctive image features from scale-invariant keypoints. *Int. Journ. of Comp. Vis.*, 20:91–110, 2003.

[17] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Trans. on Pattern Analysis & Machine Intell.*, 27(10):1615–1630, 2005.

[18] S. Munder and D. M. Gavrila. An experimental study on pedestrian classification. *IEEE Trans. on Pattern Analysis & Machine Intell.*, 28(11):1863–1868, 2006.

[19] G. Overett, L. Petersson, N. Brewer, L. Andersson, and N. Pettersson. A new pedestrian dataset for supervised learning. In *IEEE Intelligent Vehicles Symposium*, 2008.

[20] X. Ren and J. Malik. Learning a classification model for segmentation. In *Proc. of Conf. on Computer Vision*, 2003.

[21] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Trans. on Pattern Analysis & Machine Intell.*, 22:888–905, 2000.

[22] L. Spinello, R. Triebel, and R. Siegwart. Multimodal detection and tracking of pedestrians in urban environments with explicit ground plane extraction. In *IEEE Int. Conf. on Intell. Rob. and Sys. (IROS)*, 2008.

[23] L. Spinello, R. Triebel, and R. Siegwart. Multimodal people detection and tracking in crowded scenes. In *Proc. of The AAAI Conference on Artificial Intelligence*, July 2008.

[24] P. Viola, M. J. Jones, and D. Snow. Detecting pedestrians using patterns of motion and appearance. *Int. Journ. of Comp. Vis.*, 63(2):153–161, 2005.

[25] B. Wu and R. Nevatia. Detection of multiple, partially occluded humans in a single image by bayesian combination of edgelet part detectors. In *IEEE Int. Conf. on Computer Vision (ICCV)*, 2005.