2012 Intelligent Vehicles Symposium
Alcalá de Henares, Spain, June 3-7, 2012

Pedestrian Candidates Generation using Monocular Cues

Diego Cheda, Daniel Ponsa and Antonio M. López

Abstract— **Common techniques for pedestrian candidates generation (e.g., sliding window approaches) are based on an exhaustive search over the image. This implies that the number of windows produced is huge, which translates into a significant time consumption in the classification stage. In this paper, we propose a method that significantly reduces the number of windows to be considered by a classifier. Our method is a monocular one that exploits geometric and depth information available on single images. Both representations of the world are fused together to generate pedestrian candidates based on an underlying model which is focused only on objects standing vertically on the ground plane and having certain height, according with their depths on the scene. We evaluate our algorithm on a challenging dataset and demonstrate its application for pedestrian detection, where a considerable reduction in the number of candidate windows is reached.**

I. Introduction

The main objective of Advanced Driver Assistance Systems (ADAS) is increasing driver safety and comfort. ADAS systems require a full understanding of the scenarios where the vehicle is evolving, including detection of moving and stationary objects that determine the free space available for driving. In that case, the vehicle's surroundings is perceived and monitored by sensors to avoid unsafe situations (e.g., collisions). Although systems employing active sensors (e.g., radar, lidar, etc.) have shown promising results in object detection, they have several drawbacks, such as high cost, high consumption, and interference caused by sensors of the same type installed in different vehicles. However, passive sensors based on visual information (like cameras) receive a rich representation of the environment, that can be used to identify objects on the scene, as well as to detect lanes and recognize traffic signs. Indeed, due to the low cost of camera sensors, vision-based systems will be present as standard equipment on mid/low-priced vehicles providing information to ADAS applications.

A fundamental stage in scene understanding is the recognition of objects which are present in the scene (e.g., pedestrian [1], vehicles [2], signals [3]). To warn the driver in time of potential dangers, this step must be performed efficiently. Analyzing the whole image to locate potential objects locations is not feasible due to this constraint. What many object detection proposals do is follow two steps: First, hypothesize about object locations in an image, and then test this hypothesis to verify the presence of the object. Often these steps are executed multiple times on an image to recognize different objects by using independent methods.

The number of hypothesis to be evaluated can be drastically reduced by assuming that interesting objects are approximately vertical and their height is into a limited range. A reduced number of scanning windows has two advantages for an object detection module: on the one hand, speeding up the detection by discarding large image portions that not provide relevant information; on the other hand, reducing the false positives detections by focusing on specific regions with high probability of having the presence of objects.

In this paper, we propose a novel method to generate a set of candidate hypothesis based on different cues and context information which are available in single images. Our method fuses two complementary mid-level scene representations to select the image region where applying an object recognition algorithm has sense. Basically, we use geometric information obtained from a single image that allow us to distinguish between three main classes of surfaces: horizontal, vertical and those that belong to sky regions in the image [4]. From these information, we are able to know what regions are potentially supporting surfaces (i.e., the ground), and what are vertical objects. In the ADAS context, interesting objects are vertical and located over the road plane; so we have a first clue where selectively searching them.

Another useful information is the distance of objects in the scene, since it constrains the object detector's scale to be used. Traditionally, distances has been estimated using multiview approaches (e.g., stereo, structure from motion). However, this information can be obtained from a single image. We estimate coarse depth maps using a set of visual features, which are useful to determine an approximated distance of objects receding into depth. Both kind of information is combined to select regions of interest (ROI) containing possible stationary or moving vertical objects located in front of the vehicle. Figure 1 depicts our approach to pedestrian candidates generation.

Many approaches concerning our purpose have been proposed. Here, we refer some of them. For instance, popular object detection algorithms find pedestrians by exhaustively scanning over all locations and scales (i.e., a sliding window approach). Viola et al. [5] train a cascade classifier of walking humans by combining motion and appearance information. This method has good performances, but the number of windows to be evaluated is huge. Assuming a flat world, Gavrila et al. [6] generate candidates over a presupposed road plane. However, this algorithm suffers when the road is a non-flat surface and/or vehicle has pitch variations. Using stereo cameras, Badino et al. introduced a compact description of the world for autonomous vehicles, called "stixel world" [7], offering a strong simplification of the data,

Diego Cheda, Daniel Ponsa, and Antonio M. López are with the Computer Vision Center and Computer Science Department, Universitat Autònoma de Barcelona (UAB), 08193 Bellaterra, Barcelona, Spain. E-mails: {dcheda,daniel,antonio}@cvc.uab.es

978-1-4673-2118-1/$31.00 ©2012 IEEE

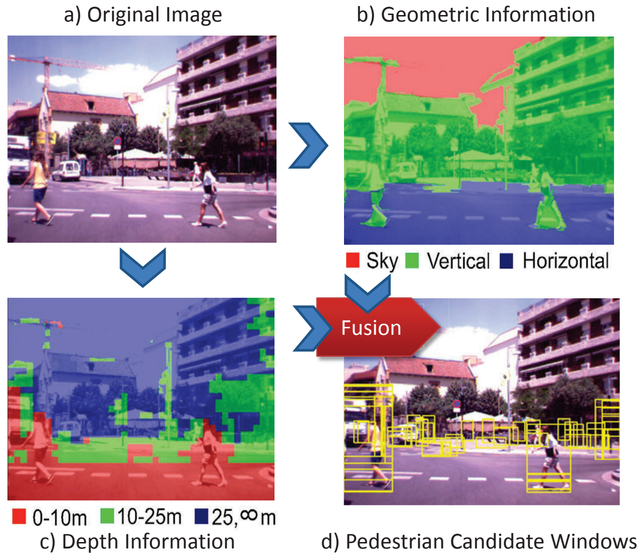


Fig. 1. Our method is based on exploiting information extracted from an image. First, pixels corresponding to vertical objects and a rough depth map are obtained. Then, we generate candidate windows from this information.

but preserving the information of interest. The underlying model focuses only on objects standing vertically on the ground plane and having certain height.

The rest of the paper is organized as follows. In the next section, we detail our method to monocularly compute the medium-level information used in our proposal. Then, we describe how we fuse both representations to generate candidate windows. Finally, we measure the performance of our approach for pedestrian candidates generation and conclude.

II. Medium-Level Representations

In this section, we describe our method to build a compact representation of a scene. Figure 1 shows a schema of our approach. Briefly, the first step is extracting useful information from the image regarding geometric and depth clues.

A. Geometric Information

Geometric information about the scene is recovered by using the approach proposed by Hoiem et al. [4]. This method segments the image into three geometric classes that depend on the orientation of the surfaces in the scene. Each region in the image is classified as horizontal, vertical or sky. Horizontal surfaces are approximately parallel to ground plane and objects can be supported by them (e.g., road surface). Vertical surfaces are roughly perpendicular to ground plane (e.g., buildings, pedestrian, cars, trees, etc.). The sky usually located on top regions in the image, corresponding to the air and clouds. Basically, an image is over-segmented into superpixels [8], each of which belongs to a particular geometric class. Each superpixel is described by depth cues, including color, location, perspective, and texture. Then, from a logistic regression form of AdaBoost previously trained, the geometric class of each superpixel is inferred. An example of the result of this process is shown in Fig. 1(b).

B. Depth Information

Even though depth estimation has been traditionally focused on techniques requiring two or more images (e.g., depth from stereo, structure from motion, etc.), recently, some proposals on depth estimation from a single image have been done [9], [10]. They try to estimate exact distances to elements in the scene, and this is achieved by a procedure requiring high computation. However, it is possible to obtain rough but valuable information of the depth of the scene with low computation methods.

Here, instead of estimating a continuous depth map, we segment the distance space into different ranges. Each range is selected taking into account the object's scale variability, whose image projection onto image plane is affected due to perspective effects.

For computing such "coarse" depth map, we train three binary Real AdaBoost classifiers [11] based on a set of discriminative features to distinguish between the different depth ranges. We use the following features: RGB and HSV color mean and histograms for each channel to distinguish between different objects, texture gradients characterized by Weibull parameters [12] and Gabor filters to capture surface orientations, and pixel location to distinguish different regions in the image (sky, ground, etc.) [13]. Each image is segmented into a regular grid of 10 × 10 pixel windows, and the feature vector is computed for each window.

To train our classifiers, we require a set of images with depth information available. In our case, we use a set of images, where each one has an associated stereo depth map. Depth maps are used to label positive and negative examples for each depth range.

Given a new image described by our set of features, the probability of belonging to each depth range is computed for each grid window by applying the previously trained classifiers. Then, we segment the image by taking for each window the maximum confidence result. An example of the result of this process is shown in Fig. 1(c).

III. Candidate Window Generation

From the previous intermediate results, we hypothesize about which vertical objects at different depths are interesting in the ADAS context. Basically, we start by dividing an image into superpixels, which is an attempt to divide the image such that boundaries coincide with image edges, grouping similar pixels into regions. Then, we combine geometric and depth information by an agglomerative hierarchical clustering [14] over the computed superpixels until the bounding box enclosing a set of superpixels has a coherent size with respect to the object size to be detected.

Our hierarchical clustering is based on the following set of physical/spatial assumptions:

- Gestalt constraints: We take into account two grouping principles of Gestalt school. On the one hand, the principle of good continuation which states that a regions which are connected have smooth boundaries. On the other hand, the principle of similarity which states that the elements in a region are similar, including similar color, brightness, and texture [15]. To fulfill the Gestalt principles, the image is over-segmented into superpixels by using Turbopixels approach [16].

- Gravity constraint: Elements in the driving environment should stand on the ground plane.
- Depth constraint: All superpixels belonging to an object are located at the same depth region, and must be grouped together.
- Size constraint: In our context, the size of an interesting object is constrained to certain range according with its depth, taking into account the camera calibration properties (i.e., focal length, and image size).

Inspired by [17], we use an agglomerative clustering method on the Euclidean distances between the coordinates of the superpixels centroids. The algorithm is composed of the following steps:

1) Start with two sets of superpixels: $\mathcal{G}$ of vertical superpixels whose distance to the ground plane is below a threshold, and $\mathcal{V}$ of the rest of vertical superpixels.
2) Find the most similar pair of superpixels, say pair (g, v), where $g \in \mathcal{G}$ and $v \in \mathcal{V}$, and the Euclidean distance $d(g, v)$ between centroids of g and v is the minimum.
3) Combine g and v to form the superpixel $g = g \cup v$ if the following conditions hold:
   a) Both g and v are located at the same depth range, and
   b) The size of the merged superpixel $g = g \cup v$ is within a certain range, according with its depth.
4) If the size of g is within the minimum and maximum sizes of an interesting object, generate a new candidate window for g.
5) Remove v from $\mathcal{V}$.
6) While g fulfills the size condition, repeat from step 2. Otherwise, select a new $g \in \mathcal{G}$ and start again from step 2, until $\mathcal{V}$ is empty.

At the end of this process, candidate windows which has certain overlapping between them can be fused to generate a new candidate as long as size and depth constraints are still satisfied.

An example of how the clustering algorithm works is shown in Fig. 2. Fig. 2(a) shows the information sources used during clustering, as we described above. In this case, we are devoted to pedestrian candidates generation, considering pedestrian's sizes for merging. In Fig. 2(b), we can observe how the superpixels are fused together into a single one as the algorithm progresses. In Fig. 2(c) each region is enclosed into a bounding box to conform a candidate window.

IV. Evaluation Results

In this section, we evaluate the performance of our algorithm for candidate windows generation with respect to state-of-the-art methods, using a public available dataset.

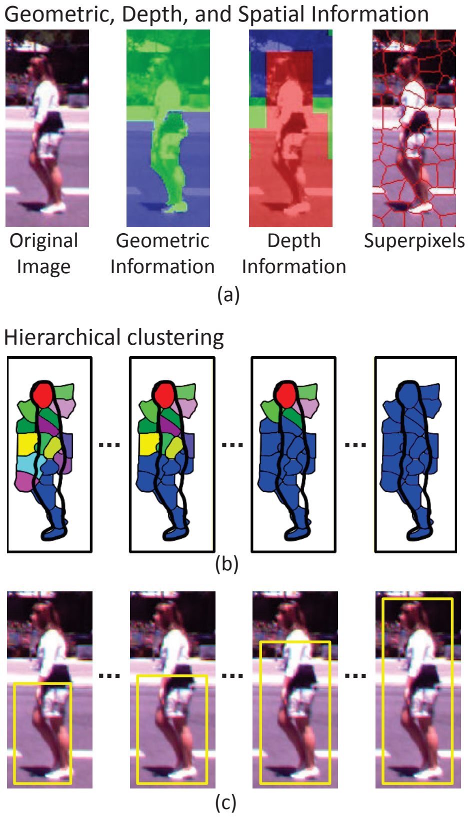


Fig. 2. Hierarchical Clustering. (a) Information sources used in our clustering algorithm, (b) An example of how superpixels are merged as the clustering algorithm progresses, and (c) Bounding boxes surrounding regions.

TABLE I
Pedestrian Sizes

Distance (m)	Minimum Size (pixels)	Maximum Size (pixels)
0 - 10	70 × 140	120 × 240
10 - 25	30 × 60	70 × 140
25 - 50	12 × 24	30 × 60

A. *Dataset and Ground Truth*

Our dataset consists of 15 sequences taken from a stereo-rig rigidly mounted in a car while it is driving on an urban scenario. Each image has an associated depth map computed from stereo images. In total, there are 4364 frames, which correspond to 7983 manual annotated pedestrians visible at less than 50 meters. This dataset is public available at `http://www.cvc.uab.es/adas/index.php?section=other_datasets`.

Table I shows the minimum and maximum size of a pedestrian at certain distance from the camera for the considered configuration. We use these pedestrian sizes as size constraints in our candidate windows generation process.

Mainly focusing on variation of pedestrians sizes along distance, we define three distance ranges: 0-10 m, 10-25 m, and more than 25 m [18]. The first two ranges are high-risk areas in case of vehicle collision against an object. The last range are a low-risk areas, where pedestrians are less vulnerable to suffer the consequences of an accident.

During the training phase of our depth-based segmentation method, we use a training set consisting of 700 images randomly taken from different sequences. The corresponding stereo depth maps are used to label a set of positive/negative examples for each distance range.

B. Evaluation Methodology

The aim of the proposed hypothesis generation is to yield few false negatives (FN, i.e. the number of missed pedestrians), while keeping the number of false positives (FP, i.e. the number of regions corresponding to non-pedestrians) low.

To judge the benefits of our approach, we compare how well the generated candidates are related to ground truth pedestrian annotations. Based on the evaluation protocol proposed by Gerónimo in [18], we measure the performance our approach in terms of the following criteria:

1) Minimizing the amount of pedestrian candidates generated PC = TP + FP.
2) Maximizing the True Positive Rate TPR $= \frac{\text{TP}}{\text{TP+FN}}$.

Each candidate window c is compared against the ground truth annotation a using the area of overlap between both bounding boxes by the formula

$$\text{overlap}(c, a) = \frac{\text{area}(a \cap c)}{\text{area}(a \cup c)} \ , \qquad (1)$$

A candidate is classified as TP, FP or FN using the overlapping measure proposed by Everingham et al. [19] for object detection evaluation in the PASCAL Challenge,

$$\text{classify}(c, a) = \begin{cases} \text{TP} & \text{if overlap}(c, a) > \Gamma \\ \text{FP} & \text{if overlap}(c, a) \leq \Gamma \\ \text{FN} & \text{if } a \text{ does not have any associated} \\ & \text{candidate } c \ . \end{cases} \qquad (2)$$

In our case, for a candidate c to be a TP, we require that this overlap exceeds a threshold $\Gamma = 50\%$.

C. Performance Evaluation

In the literature, there are several methods to generate pedestrian candidates. Here, we only briefly describe the most relevant strategies for comparing them with our method in terms of performance. A detailed description of the following strategies can be found in [18].

The simplest candidate generation method for pedestrian detection is the sliding window approach [20] which is an exhaustive scan over the input image with windows of different scales at all the possible positions. The drawback of this approach is that requires generating a big number of candidates to reach an acceptable performance. A big number of candidates implies a higher computation time during classification, which is undesirable. Additionally, many of these candidates are false positives since this method does not use any prior knowledge.

Other technique is based on the so-called flat world assumption [21]. In this case, pedestrians are assumed to be on a planar road. This is a strong constraint which implies that the road geometry and its position with respect to the camera is known and remains constant along time. Under these conditions, the algorithm generates candidates over the presupposed road plane with pedestrian-sized windows.

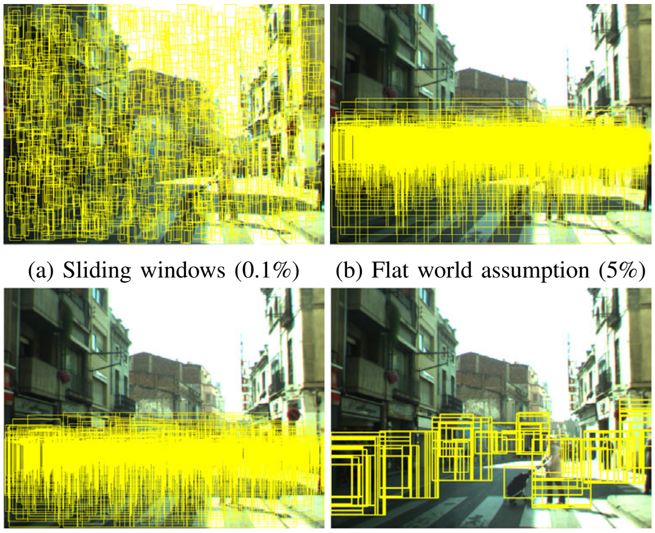

(a) Sliding windows (0.1%) (b) Flat world assumption (5%)
(c) Adaptive road scanning (5%) (d) Our approach (100%)

Fig. 3. Qualitative evaluation. Here, we can see the candidate windows generated by the considered approaches versus our results. The number in parenthesis indicates the percentage of displayed windows. The number of candidate windows is significantly reduced by applying our approach. Figures (a)-(c) were taken from [18].

However, due to road imperfections, car accelerations, and changes in the road slope, the camera pose changes, and the image is scanned sub-optimally. Then, road geometry and camera pose cannot be assumed as constant. The limitations of flat world assumption-based method can be overcome by adjusting the scanning grid to a road surface estimated dynamically, as Gerónimo proposes [18]. The algorithm estimates the road surface based on 3D points provided by a stereo camera, and then performs a road scanning in the same way that the previous method.

Figure 3 depicts a qualitative comparison between our proposal and the methods described above. We can observe that our approach selects a reduced number of candidate windows with respect to the rest of considered methods. Table II shows the results in terms of TPR and PC per frame of each algorithm.

Although sliding window has the best performance with respect to TPR, the number of candidates to classify is big, which affects the time consumption in a posterior classification stage.

Assuming a flat world the search space is significantly reduced, but the TPR is low. This implies that many pedestrians will be lost during this process. The TPR drops due to the camera motion (mainly, pitch angle variations) produced by road slopes, which produce that in many cases the fixed plane does not coincide with the real one, and hence the generated candidate windows are not correct.

A trade-off between TPR and the number of candidates is reached using adaptive road scanning. However, the TPR is

TABLE II

COMPARISON BETWEEN APPROACHES

Algorithm	PC	TPR
Sliding Window (perfect)	1.300.000	100%
Sliding Window (dense)	700.000	98%
Sliding Window (sparse)	220.000	75%
Flat World Assumption	42.000	35%
Adaptive Road Scanning	32.000	74%
Our approach	500	84%

not perfect.

Still far from being perfect, our method reduces remarkably the search space but, at same time, maintains a high performance with respect to TPR. The obtained reduction in the number of candidates is very significantly since we combine strong clues about the physical world for filtering the search space. The used priors regarding vertical surfaces and its depths, coupled with our spatial restrictions, allow us to focus on image regions where the probability of having pedestrians is relatively high.

To reach a similar performance to our method, sliding window approach requires generating approximately 300.000 windows, which is 600 times more than the candidates generated by our method.

Figure 4 shows qualitative results obtained with our method. Figures 4(b) and (c) show geometric and depth information used in our candidates generation process. As we depict in Fig. 4(d), the windows are posed only over interesting regions for our context, while large image portions are discarded.

Figure 5 shows an histogram of the number of pedestrians not included in the hypothesis generated by our process. We can see that lost pedestrians are mainly located at far distances from the car. This is because the far pedestrians have smaller sizes (i.e., very few pixels) due to the sensor resolution. Then, they are hard to be segmented into their constituents parts by the superpixel algorithm and to agglomerate by our approach. However, these pedestrians are outside the high-risk area, and will be further detected with very high probability when the car approaches to them, since our proposal includes 99.4% of pedestrian in the range 0-10 m. From our opinion, this candidate distribution with respect to distances is preferable since closer pedestrians are the most vulnerable ones and require special efforts.

V. CONCLUSIONS

In this paper, we have presented a novel monocular method for generating pedestrian candidates. Our method is based on cues which involve two relevant sources of information about a scene: geometric relationships and depth. Geometric information is extracted by using the approach proposed by Hoiem et al. [4], whereas depth is roughly computed by a multiclass classifier approach. Both clues are combined to generate pedestrian candidates by a hierarchical clustering. This clustering agglomerates pixels which are related through physical properties like appearance, gravity, proximity, and size constraints that we impose.

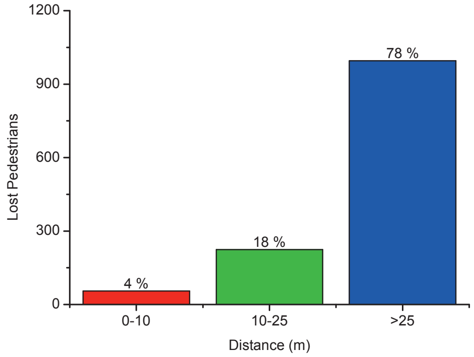


Fig. 5. Histogram of pedestrians which are lost during our process. We can see that mainly they are pedestrian located at far distances from the car, and they will be further detected.

We have evaluated our model for pedestrian candidates generation and compared it with respect to other approaches in the state-of-the-art. The results shown that our method is useful for that task since significantly reduces the number of candidates to be evaluated by a posterior pedestrian classifier, loosing few pedestrians as the high value for TPR shows.

As future work, we plan to integrate our method in a pedestrian detection system to evaluate how it benefits the overall performance of such system. We also are interested in improving pedestrian classifiers by using depth information. Due to the high variability of the pedestrian at different distances, we are planning to train different classifier models depending on the target distance provided by our approach to compute rough depth maps.

ACKNOWLEDGMENTS

This work is supported by Spanish MICINN projects TRA2011-29454-C03-01, TIN2011-29494-C03-02, Consolider Ingenio 2010: MIPRCV (CSD200700018), and Universitat Autònoma de Barcelona.

REFERENCES

[1] D. Gerónimo, A. López, A. Sappa, and T. Graf, "Survey of Pedestrian Detection for Advanced Driver Assistance Systems," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 7, pp. 1239 –1258, 2010.

[2] D. Ponsa, J. Serrat, and A. M. López, "On-board Image-based Vehicle Detection and Tracking," *Trans. Inst. Meas. Control*, vol. 33, pp. 783–805, 2009.

[3] R. Timofte, K. Zimmermann, and L. Van Gool, "Multi-view Traffic Sign Detection, Recognition, and 3D Localisation," in *IEEE Workshop App. Comput. Vision*, 2009, pp. 1–8.

[4] D. Hoiem, A. Efros, and M. Hebert, "Recovering Surface Layout from an Image," *Int. J. Comput. Vision*, vol. 75, no. 1, pp. 151–172, 2007.

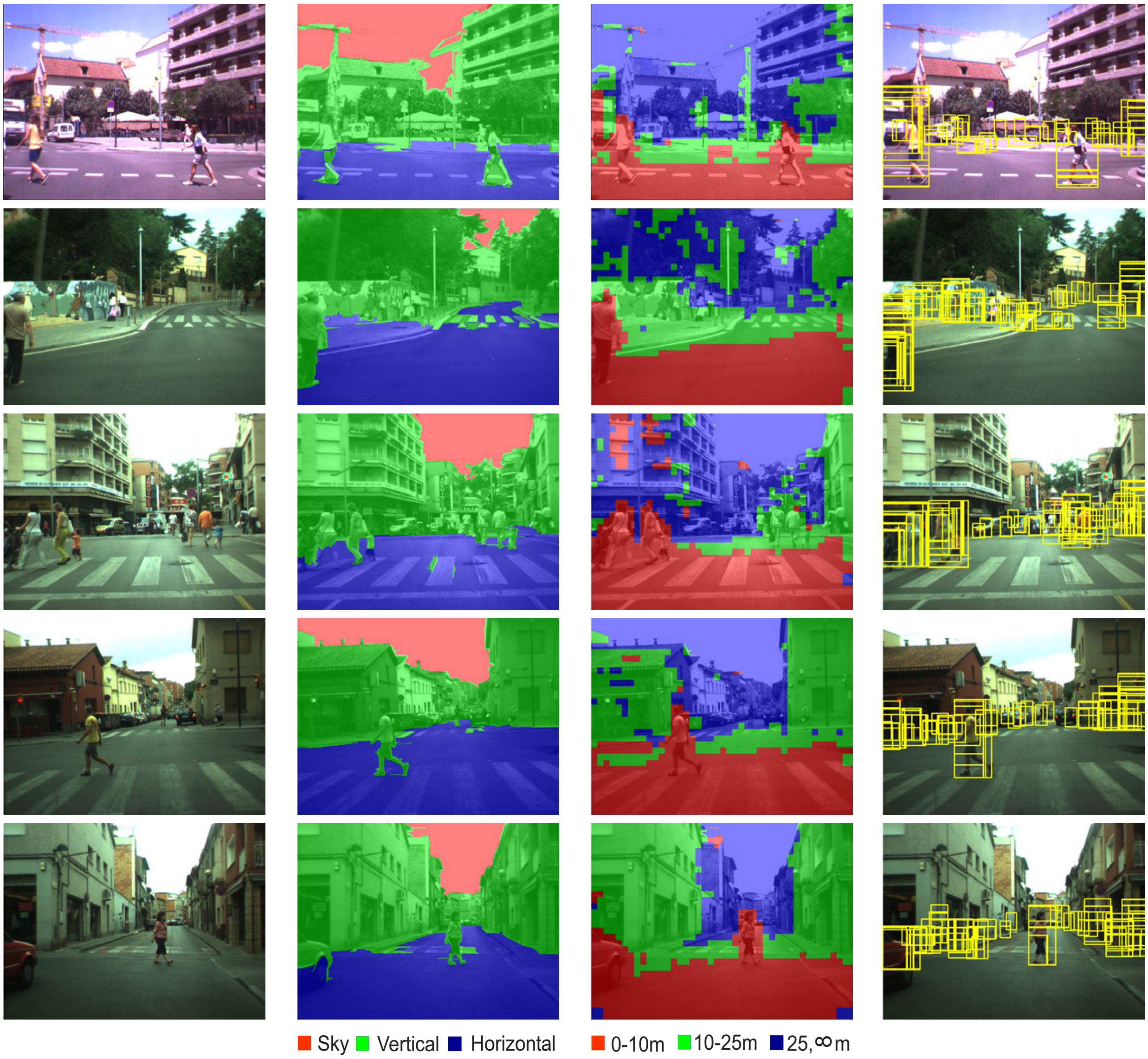
[5] P. Viola, M. J. Jones, and D. Snow, "Detecting Pedestrians Using Patterns of Motion and Appearance," *Int. J. Comput. Vision*, vol. 63, pp. 153–161, 2005.

[6] D. Gavrila and S. Munder, "Multi-cue Pedestrian Detection and Tracking from a Moving Vehicle," *Int. J. Comput. Vision*, vol. 73, pp. 41–59, 2007.

[7] H. Badino, U. Franke, and D. Pfeiffer, "The Stixel World - A Compact Medium Level Representation of the 3D-World," in *DAGM Symp. Pattern Recognit.*, 2009, pp. 51–60.

[8] X. Ren and J. Malik, "Learning a Classification Model for Segmentation," in *IEEE Int. Conf. Comput. Vision*, 2003, pp. 10–17.

[9] A. Saxena, M. Sun, and A. Ng, "Make3D: Learning 3D Scene Structure from a Single Still Image," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 5, pp. 824–840, 2009.



(a) Original image (b) Geometric information (c) Depth information (d) Candidate windows

Fig. 4. Results of our approach to build a compact representation of the world: (a) Original image, (b) Geometric information, (c) Depth information, and (d) Candidate windows.

[10] B. Liu, S. Gould, and D. Koller, "Single Image Depth Estimation from Predicted Semantic Labels," in *IEEE Conf. Comput. Vision Pattern Recognit.*, 2010, pp. 1253–1260.

[11] R. Schapire and Y. Singer, "Improved Boosting Algorithms Using Confidence-rated Predictions," *Mach. Learn.*, vol. 37, no. 3, pp. 297–336, 1999.

[12] J. Geusebroek and A. Smeulders, "A Six-Stimulus Theory for Stochastic Texture," *Int. J. Comput. Vision*, vol. 62, pp. 7–16, 2005.

[13] D. Hoiem, A. Efros, and M. Hebert, "Automatic Photo Pop-Up," *ACM Trans. on Graphics*, vol. 24, no. 3, pp. 577–584, 2005.

[14] R. Sibson, "SLINK: An Optimally Efficient Algorithm for the Single-Link Cluster Method," *Comput. J.*, vol. 16, no. 1, pp. 30–34, 1973.

[15] B. Goldstein, *Sensation and Perception*. Belmont, California, USA: Wadsworth Cengage Learning, 2010.

[16] A. Levinshtein, A. Stere, K. N. Kutulakos, D. J. Fleet, S. J. Dickinson, and K. Siddiqi, "TurboPixels: Fast Superpixels Using Geometric Flows," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, pp. 2290–2297, December 2009.

[17] K. van de Sande, J. Uijlings, T. Gevers, and A. Smeulders, "Segmentation as Selective Search for Object Recognition," in *IEEE Int. Conf. Comput. Vision*, 2011, pp. 1879–1886.

[18] D. Gerónimo, "A Global Approach to Vision-based Pedestrian Detection for Advanced Driver Assistance Systems," Ph.D. dissertation, Computer Vision Center, Universitat Autnoma de Barcelona, 2010.

[19] M. Everingham, A. Zisserman, C. K. I. Williams, L. V. Gool, M. Allan, C. M. Bishop, O. Chapelle, N. Dalal, T. Deselaers, G. Dork, S. Duffner, J. Eichhorn, J. D. R. Farquhar, M. Fritz, C. Garcia, T. Griffiths, F. Jurie, D. Keysers, M. Koskela, J. Laaksonen, D. Larlus, B. Leibe, H. Meng, H. Ney, B. Schiele, C. Schmid, E. Seemann, J. Shawe-taylor, A. Storkey, O. Szedmak, B. Triggs, I. Ulusoy, V. Viitaniemi, and J. Zhang, "The 2005 PASCAL Visual Object Classes Challenge," in *First PASCAL Challenges Workshop*, 2006.

[20] N. Dalal, "Finding People in Images and Videos," Ph.D. dissertation, Institut National Polytechnique de Grenoble, 2006.

[21] M. Bertozzi, A. Broggi, R. Chapuis, F. Chausse, A. Fascioli, and A. Tibaldi, "Shape-Based Pedestrian Detection and Localization," in *IEEE Conf. Intell. Transp. Syst.*, 2003, pp. 328–333.