

Fusing LIDAR, camera and semantic information: A context-based approach for pedestrian detection

Cristiano Premebida and Urbano Nunes

Abstract

In this work, a context-based multisensor system, applied to pedestrian detection in urban environments, is presented. The proposed system comprises three main processing modules: (i) a LIDAR-based module acting as the primary object detector; (ii) a module which supplies the system with contextual information obtained from a semantic map of the roads, and (iii) an image-based detection module, using sliding window detectors, with the role of validating the presence of pedestrians in the regions of interest generated by the LIDAR module. A Bayesian strategy is used to combine information from sensors onboard the vehicle ('local' information) with information contained in a digital map of the roads ('global' information). To support experimental analysis, a multisensor dataset, named the Laser and Image Pedestrian Detection dataset (LIPD), is used. The LIPD dataset was collected in an urban environment, under daylight conditions, using an electrical vehicle driven at low speed. A down-sampling method, using support vectors extracted from multiple linear SVMs, was used to reduce the cardinality of the training set and, as a consequence, to decrease the CPU time during the training process of the image-based classifiers. The performance of the system is evaluated, in terms of detection rate and the number of false positives per frame, using three image-detectors: a linear SVM, a SVM-cascade, and a benchmark method. Additionally, experiments are performed to assess the impact of contextual information on the performance of the detection system.

Keywords

sensor fusion, sensing and perception, computer vision, recognition, field robots, field and service robotics, range sensing

1. Introduction

Multisensor data fusion plays an important role in the fields of Intelligent Transportation Systems (ITS) and Intelligent Vehicles (IV), as evidenced by the significant amount of work carried out in these areas, see for instance the recent surveys of Geronimo et al. (2010), Faouzi et al. (2011), Stiller et al. (2011) and Dollar et al. (2012). Advanced Driver Assistance Systems (ADAS) have also deserved much attention in recent years because of their many applications in the automotive market e.g. Adaptive Cruise Control (ACC), Lane Departure Warning (LDW), Anti-lock Braking Systems (ABS), Collision Warning Systems (CWS), and Pedestrian Protection Systems (PPS). The latter, which is the focus of this work, can be divided, in general terms, into two fields of research: passive and active systems, as mentioned by Gandhi and Trivedi (2007). The former is characterized by built-in safety features on vehicles, designed primarily to mitigate possible injuries to pedestrians due to impact e.g. specially designed front bumpers, deformable hoods, specific airbags placed near the frontal columns of the vehicle, and so on. Whereas, active pedestrian protection systems are based on sensors

onboard the vehicle, and/or on the infrastructure, with the role of predicting and anticipating the possible risk of collisions.

Onboard sensor-based pedestrian detection systems (Gandhi and Trivedi, 2007), in the domain of ADAS applications, is a research topic which has received considerable attention, as evidenced by the work of Felzenszwalb et al. (2010), Markoff (2010), Navarro-Serment et al. (2010), Spinello et al. (2010), Douillard et al. (2011) and Garcia et al. (2011). In particular, context-based perception systems use contextual cues extracted from still frames e.g. scene attributes and spatial relations between objects as proposed by Perko and Leonardis (2010), or contextual information like scale, distance, and road location, which can be

Institute of Systems and Robotics, University of Coimbra, Portugal

Corresponding author:

Cristiano Premebida, University of Coimbra, Polo II Pinhal do Marrocos
Coimbra, Coimbra 3030-290, Portugal.

Email: cpremebida@isr.uc.pt

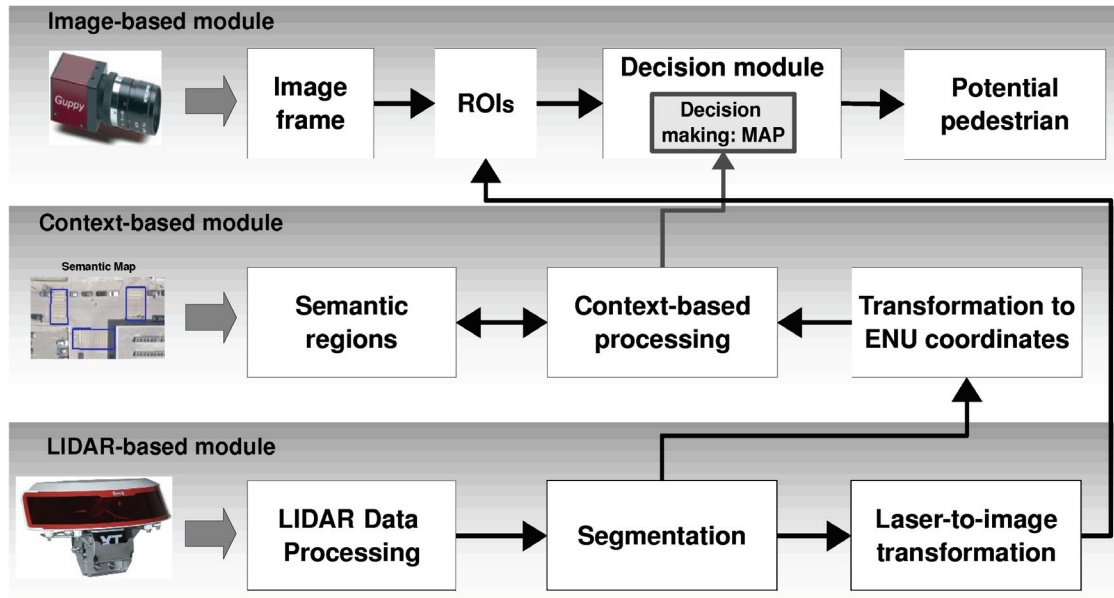


Fig. 1. Block diagram illustrating the main processing modules comprising the pedestrian detection system with contextual information incorporated into the system.

obtained by LIDARs or stereovision as discussed by Geronimo et al. (2010). The aforementioned work shares a common element: the information is extracted from onboard sensors. On the other hand, solutions for pedestrian detection combining ‘external’ sources of contextual information e.g. GPS-based semantic maps, seem to be an interesting approach that has rarely, if ever, been addressed by the IV/ITS community.

Information from image-based detectors (‘local’ information) is combined with information contained in the semantic map (‘global’ information) by means of a Maximum A Posteriori (MAP) strategy. More specifically, the confidence scores of the image detectors are obtained from the classifier outputs which are handled as conditional probabilities, while the information obtained from the map enters into the system in the form of prior probabilities. Finally, the MAP decision module outputs the set of detection windows with their a posteriori confidence score.

In this work, we propose an active pedestrian detection system which combines data from a LIDAR and a monocular camera mounted onboard an electrical vehicle, with information obtained from a semantic map of the roads where the vehicle was driven. Some regions of this map were labeled as regions where the presence of potential pedestrians is more ‘likely’ to occur e.g. cross-walks and bus stops. The multisensor information is processed using an architecture which comprises three main processing modules: (i) a LIDAR-based module; (ii) a context-based module; and (iii) an image-based module. The LIDAR-based module is in charge of primary object detection i.e. it generates a set of detected objects in the form of laser

segments, or clusters, which are transformed to a local navigation system where the positions of the objects in the map are used as contextual information. Since the laser is calibrated w.r.t. the camera, the position of the set of objects is used to project a set of regions of interest (ROIs) in the image plane. Inside each ROI, an image-based classifier is used in the form of multiscale sliding window detectors which are shifted in position and size for searching for pedestrian evidence. These sliding window detectors, or simply detection windows, are characterized by spatial parameters, in pixel coordinates, and by a confidence score given by a classification method.

Concerning the image-based module, our proposed SVM-cascade method, succinctly presented by Ludwig et al. (2011), is compared in terms of detection performance with a single linear Support Vector Machine (SVM) and with the deformable parts-based models proposed by Felzenszwalb et al. (2010) and available at <http://cs.brown.edu/~pff/latent-release4/>. Additionally, we propose a down-sampling method, using support vectors extracted from multiple linear SVMs, to reduce the cardinality of the training set and to decrease the CPU time during the training process of the classifiers.

In summary, the proposed system’s architecture is illustrated in Figure 1. The LIDAR-based module and its processing stages are described in Section 2, while the image-based module, the down-sampling approach and the SVM-cascade are detailed in Section 3. The context-based module, using a map of context-regions of the scenario, is presented in Section 4. Experiments in pedestrian detection, using the Laser and Image Pedestrian Detection (LIPD)

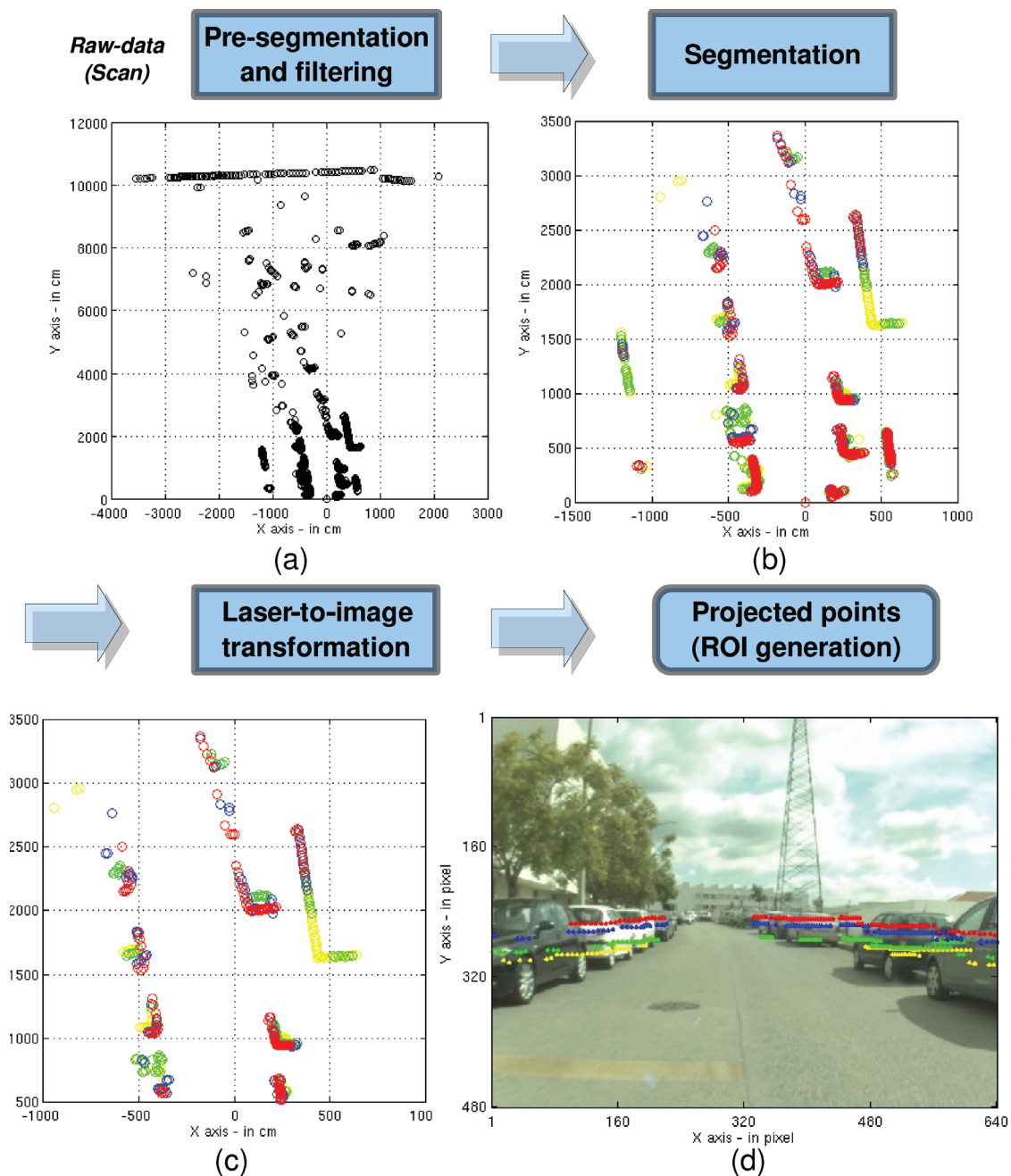


Fig. 2. Main processing steps in the LIDAR-based module. (a) Raw range points in Cartesian coordinates. (b) Points are filtered and grouped per layer, where each color indicates a given layer. (c) Range points outside the camera FOV are discarded. (d) Projection of the points, per layer, on the image frame.

dataset, are reported in Section 5. Finally, Section 6 presents the conclusions.

2. The LIDAR-based module

Our LIDAR-based system acts as the primary object detector, where each detected object constitutes a hypothesis of being a positive (pedestrian) or a negative (non-pedestrian). This module outputs a set of laser segments, henceforth

called segments, that are transformed into image coordinates and projected on the image frame in the form of ROIs, as depicted in Figure 2. Concisely, the main processing modules performed in the LIDAR-based module are:

1. **Pre-segmentation and filtering**: this module comprehends a set of pertinent data-processing tasks, necessary to decrease the complexity and processing time of subsequent stages, such as: filtering out isolated/spurious

range points, discarding measurements that occur outside a predefined Field of View (FOV), and data alignment.

2. LIDAR segmentation: this module outputs a set of segments obtained by a range data segmentation method.
3. Transformation to ENU coordinates: segments are transformed from the 2D Cartesian coordinates to a 'local' navigation frame, defined by the east, north and up (ENU) reference system, as detailed by Drake (2002).
4. Laser-to-image transformation: defined as the set of rigid coordinate transformations necessary to project the segments into the image plane. This module outputs the set of ROIs.

Figure 2 illustrates the spatial evolution of the laser data throughout the processing stages of our LIDAR-based module. This module receives, in each iteration step, a raw scan of laser points which are processed towards ROI generation on the image frame. The color of the laser points follows the standard convention adopted by the laser scanner manufacturer.

The laser and the monocular camera were calibrated according to the following steps: (1) a calibration dataset, with synchronized laser scans and image frames, was collected. The platform was held stationary, while the checkerboard was positioned 2–7 m away from the platform; (2) the intrinsic and extrinsic parameters of the camera were estimated using the calibration toolbox (http://www.vision.caltech.edu/bouguetj/calib_doc/); (3) finally, the extrinsic parameters of the camera w.r.t. the laser were obtained using the method proposed by Vasconcelos et al. (2012). These parameters were assumed to be constant during the experimental dataset collection. Points in the camera reference system $P^C = [P_X^C, P_Y^C, P_Z^C]$ can be transformed into the laser coordinate system $P^L = [P_X^L, P_Y^L, P_Z^L]$ using the transformation $P^L = (R_C^L P^C + T_C^L)$, where R_C^L is the 3×3 orthonormal rotation matrix representing the camera's orientation relative to the laser and T_C^L is the three-dimensional vector representing the relative position. The method described by Vasconcelos et al. (2012) was used to estimate R_C^L and T_C^L . The transformation between a point in the laser coordinate system P^L to a point in the camera reference system P^C is obtained by $P^C = (P^L - T_C^L) / R_C^L$. The 3D point P^C is normalized and the distortion coefficients are applied in order to obtain X_N , as described by Heikkilä and Silven (1997). Finally, the pixel coordinates in the image plane are calculated as follows.

Denoting a point in the image plane by $P^I = [u, v]$, where u and v are pixel coordinates, and considering a pinhole model, the coordinates of P^I are calculated:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K \begin{bmatrix} X_N(1) \\ X_N(2) \\ 1 \end{bmatrix} \quad (1)$$

with the camera matrix K given by

$$K = \begin{bmatrix} f_c(1) & \alpha_c f_c(2) & cc(1) \\ 0 & f_c(2) & cc(2) \\ 0 & 0 & 1 \end{bmatrix} \quad (2)$$

where f_c is the focal length, cc is the principal point, and α_c is the skew coefficient.

With the LIDAR data it is only possible to obtain the horizontal limits of the object position on the image. If it is assumed that the vehicle moves on a "flat" surface, and knowing the distance from the laser to the ground, it is easy to calculate the bottom limit of the ROI. The top limit of the ROI was estimated using the distance to the object and considering 2.5 m as the maximum height of a pedestrian. The following matrix, necessary to ensure rigid correspondence between the laser scanner and the camera reference system, was obtained:

$$[R_C^L \quad T_C^L] = \begin{bmatrix} 0.99986 & -0.014149 & -0.0093947 & 11.917 \\ 0.014395 & 0.99954 & 0.026672 & -161.26 \\ 0.009013 & -0.026804 & 0.9996 & 0.77955 \end{bmatrix} \quad (3)$$

where the translational vector components are in mm.

3. The image-based module

The pedestrian detection system involves a number of spatio-temporal processing techniques which aim to obtain, on the image frame, the estimated position and the size (scale) of potential pedestrians. One of the key problems in monocular image-based pedestrian detection is the huge number of negatives (potential false alarms) in contrast with the number of positives, demanding vast processing time and a high confidence detector. The approach adopted to deal with this problem was to use a LIDAR to generate a set of ROIs on the image frame. Inside each ROI, a image-based classifier is used in the form of a detection window which is shifted in position and size for searching for pedestrian evidence. This approach decreases the computational processing time, restricting the zones of interest on the image to a dozen ROIs at most, and reducing the number of false positives.

The image-based system discussed in this work is composed using an image-based classification method, in the form of a detection window, followed by a non-maximum suppression filtering and a decision-making stage:

1. Detection window: an image-based classifier, in the form of a multiscale sliding window detector, is the primary stage used to identify potential pedestrians inside the ROIs.
2. Non-maximum suppression: a pairwise non-maximum suppression technique is used to discard the less confident detector, of every pair of detection windows, that spatially overlap a region on the image.

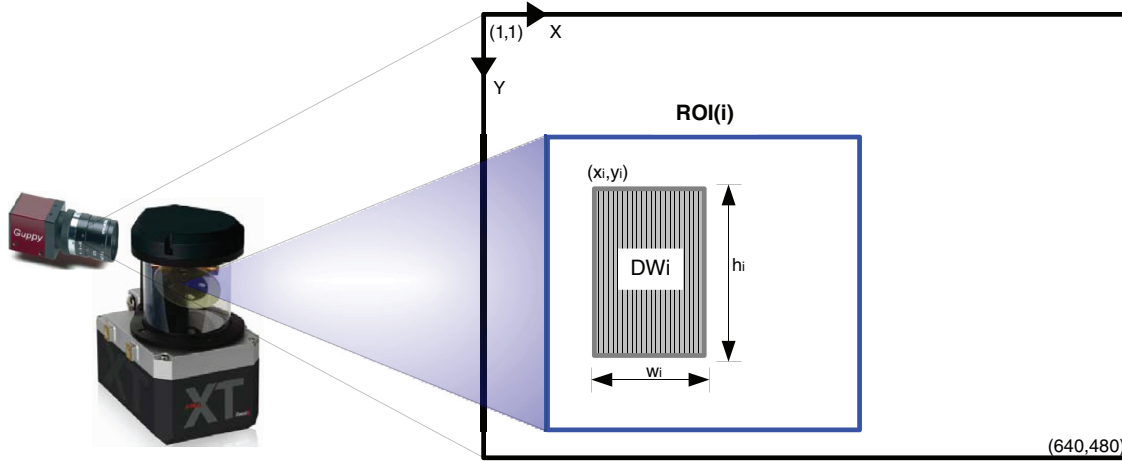


Fig. 3. Representation of a detection window, DW_i , with size defined by (w_i, h_i) , which is shifted, inside a ROI, at locations given by (x_i, y_i) .

3. Decision making: a MAP decision rule, integrating the confidence scores and contextual priors, is used to decide the presence of a potential pedestrian on the image.

Three image-based classification methods are used in this work: a linear SVM, the proposed SVM-cascade, and the deformable parts-based model. The models were learned using our training set composed of HOG¹ descriptors, and in the first two methods we used the codes made available by Dollar et al. (2012) to extract HOG features. A detection window, denoted by DW_i , is used under a multiscale sliding approach thus, DW_i is shifted inside each ROI by varying the location (x_i, y_i) and the size (w_i, h_i) as a function of a set of spatial stride steps and scale factors; see Figure 3. A given detector $DW_i = [px_i, py_i, w_i, h_i, \alpha_i]$ is defined by a rectangular area $\mathcal{A}_i$ in the ROI, with position $[px_i, py_i]$ and with size given by w_i (width) and h_i (height) in pixel coordinates, and by a confidence value α_i which is the output of the classifier associated with DW_i .

An inevitable problem that arises in the multiscale sliding window approach is the occurrence of multiple detection windows in the same ‘neighborhood’ area in the ROI. To solve this problem, a non-maximum suppression method, inspired in the methods of Dollar et al. (2009) and Enzweiler and Gavrila (2009), was used to discard multiple-detector occurrences around close/similar locations. The ratio γ between the intersection and the union area of overlapping detection windows is calculated and, for $\gamma > 0.9$ the detector with the greatest confidence score is retained and the remainders are discarded. The subsequent, and final, processing stage involved in the image-based module is decision making. In the usual case, a threshold is used for deciding the class of an observed feature vector. However, when context is available, the classification decision takes the form of a MAP rule, as described in Section 4.

It is important to underline a specific although critical characteristic of realistic (with large cardinality) datasets

used for pedestrian detection in urban scenarios: the large difference between the number of positive and negative samples. To give a clear insight about this, let's consider the size (width and height in pixels) of a pedestrian detector as $\Delta_i = s_i(w, h) \mid i = 1, \dots, 11$, where $h = 2w$, and with scales given by $s = (1.0, 1.2, \dots, 2.8, 3.0)$. The detection windows are shifted through the image at horizontal and vertical locations defined by $s_i w/4$ and $s_i h/4$ respectively. For the case of an image with 640x480 pixels, and considering $w = 27$, it generates, depending the approximations done to keep Δ with integer pixel values, at least 12,000 detection windows per frame. In a dataset with 10,000 frames, and at an average rate of 0.5 pedestrians per frame (which is a realistic factor in a typical urban scenario), it gives at least 1.2×10^8 negative detection windows against 5000 positives cutouts.

To avoid bias problems and infeasible computational requirements in such large unbalanced datasets, a down-sampling algorithm is desirable, as mentioned by Kang and Cho (2006). To preserve the information which is relevant in computing the classifier-separating hyperplane and, at the same time, to reduce the training set cardinality and decrease the complexity of the training process, a SVM-based data selection, which is inspired in the parallel-SVM architecture introduced by Graf et al. (2005), is proposed. This method, which can also be used to obtain a balanced or soft-unbalanced dataset, is discussed in the sequel.

3.1. SVM-Based down-sampling algorithm

The notations used to explain the SVM-based data selection algorithm are:

$X_P = \{X_{P(i)} : i \in I_p = \{1, \dots, np\}\}$ is the input set of positive training examples.

$X_P^S = \{X_{P(i)}^S : i \in S_p \subset I_p\}$ is a subset of selected positive training examples.

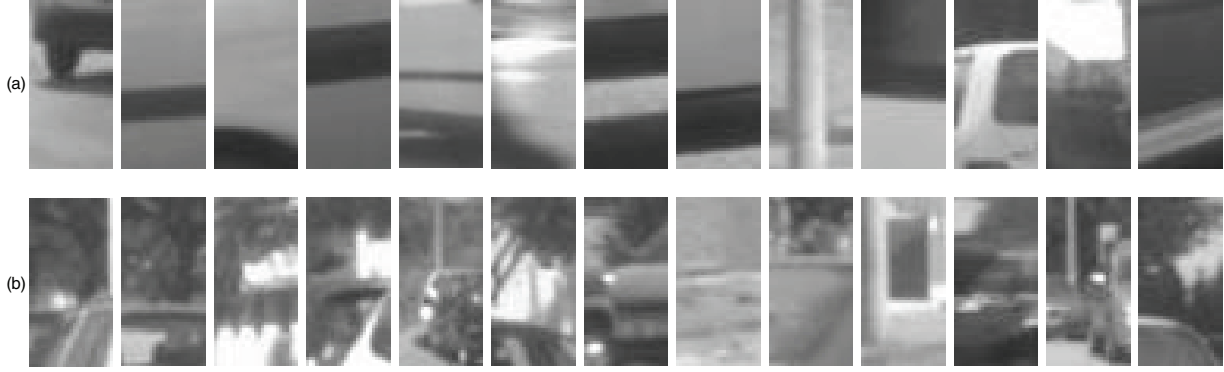


Fig. 4. Negative samples having feature vectors positioned ‘far’ from the separation margin (a); and samples which correspond to support vectors (b).

$X_N = \{X_{N(i)} : i \in In = \{1, \dots, nn\}\}$ is the input set of negative training examples.

$X_N^S = \{X_{N(i)}^S : i \in Sn \subset In\}$ is a subset of selected negative training examples.

$X_{P,N} = \{X_P, X_N\}$ is the complete training set composed of X_P and X_N .

$X_{P,N}^S = \{X_P^S, X_N^S\}$ is the selected training set.

$X_{N-\Omega} = \{X_{N(i)} : i \in In \setminus \Omega\}$ is the subset of negative instances X_N with those in Ω removed, where $\Omega \subset X_N$.

The number of negative instances nn in a realistic dataset heavily outnumbers the positive instances np , hence $np \ll nn$. For this reason, the down-sampling algorithm applied in this work selects, from the negative training set X_N , a subset of instances X_N^S with $|Sn| < |In|$. Given the initial training set $X_{P,N}$, with np positive and nn negative training examples, this *under-sampling* algorithm selects ns instances which correspond to the support vectors of X_N , denoted by SV . The first step of the down-sampling algorithm is to split X_N on n subsets $\Omega_i \subset X_N, i = 1, \dots, n$; further, for each subset Ω_i , an SVM classifier is used to extract the support vectors which will be used to compose SV . Thus, each i^{th} SVM is trained with a subset comprising np positives and $\frac{nn}{n}$ negatives. The final step is to aggregate all the negative support vectors obtained from the n SVMs. Lastly, this method outputs the selected negative training set which is composed of the negative support vectors: $X_N^S \leftarrow SV$ (see Algorithm 1). Figure 4(a) illustrates some negative samples whose training feature vectors do not correspond to support vectors and (b) illustrates some samples which correspond to support vectors.

We performed experiments on a validation set to assess the performance of the proposed down-sampling approach in terms of classification. A linear SVM and the Fisher’s Linear Discriminant classifier were trained with (i) all examples of a training set, having 100,000 negative examples, and with (ii) a reduced subset consisting of the support vectors (SV) obtained from the former training set; thus, two models have been learned. We then applied the classifiers on a testing set and, on average, both classifiers

obtained a similar hit rate but a reduction of the order of 10% in the number of false positives was observed on the models learned with support vectors.

Algorithm 1 SVM-Based down-sampling algorithm for negative example selection

Input: $X_{P,N} = \{X_P, X_N\}$: training set;

Output: X_N^S : set of selected negative samples;

- 1: $\{SV\}_i$: set of support vectors; n : number of iterations;
 - 2: Ω_i : subset of X_N ;
// Initialization:
 - 3: $X_N^S \leftarrow \{\}$: empty set;
 - 4: Decompose X_N on n subsets Ω_i , where $\Omega_i \subset X_N$;
// Selecting samples:
 - 5: **for** $i = 1; i < n; i+1$ **do**
 - 6: Train a linear SVM classifier using $\{X_P, \Omega_i\}$;
 - 7: Use the ‘negative’ support vectors from Ω_i , to generate $\{SV\}_i$;
 - 8: **end for**
// Composing X_N^S :
 - 9: $X_N^S \leftarrow \{SV\}_i, i = 1, \dots, n$.
-

3.2. The SVM-cascade method

The proposed SVM-cascade is trained using a boosting process where the number of features, in a given stage, increases w.r.t. to the preceding stage; thus, the complexity of the cascade and its classification capability increases as more stages are added to the structure. The SVM-cascade is a chain of linear SVMs, which eliminates negatives in each stage as proposed by Viola and Jones (2001); thus, at each stage the instances classified as negatives are eliminated and, conversely, the positives follow through the cascade structure as illustrated in Figure 5. Each stage of the cascade is trained to classify a given true positive rate (TP), by adjusting the separating hyperplane of the SVM in the current stage, while rejecting the negatives correctly classified.

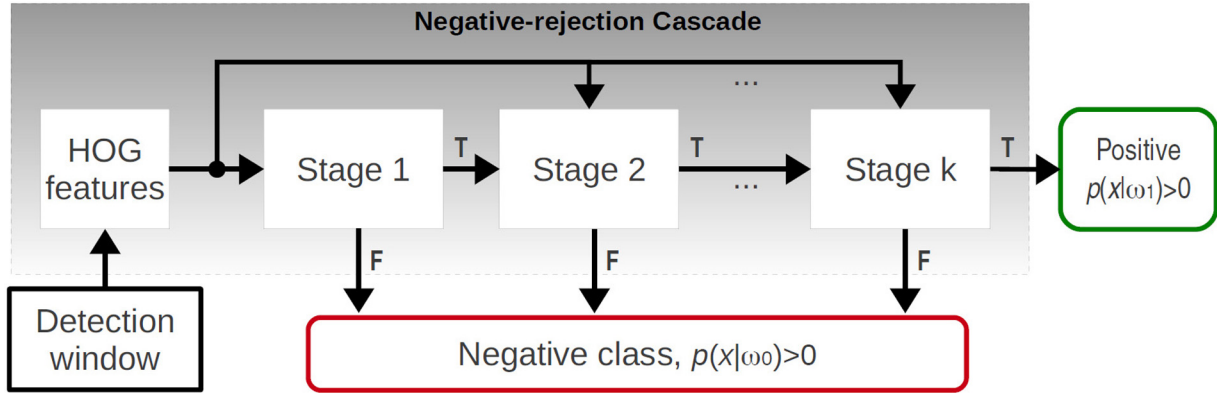


Fig. 5. Functional diagram illustrating the SVM-cascade detector: each stage rejects the samples classified as negative (F: false), while the positives (T: true) pass through all the stages to finally be detected as a pedestrian.

This process is performed by varying the bias (threshold) of the component SVM until TP is achieved. The subsequent stage of the cascade receives all the positives and the false-negative instances from the previous stage, consequently, the training set becomes more and more difficult to classify. To improve the classification capability, the number of features is incremented progressively as the number of stages increases; that is, the number of features (and the complexity) of a given stage is increased by adding n_f features w.r.t. the previous stage.

The feature vector used to train the SVM-cascade, and a linear SVM, is composed of 396 elements, defined by a normalized HOG descriptor with 4×11 cells and 9 bins. Each stage of the SVM-cascade was trained using 1 bin of the grid, totalling a cascade with 9 stages, each one with 44 features. Thus, the first stage has 44 features (first bin), the second 88 (first and second bins), and so on. The training method is summarized in Algorithm 2.

4. The context-based module

Context, in the domain of pattern recognition, represents input-dependent information, other than from the object pattern itself, used to improve the classification or detection performance; a general definition of context is provided by Duda et al. (2001), while a description of image-based contextual cues is given by Perko and Leonardis (2010). Here, context refers to the location of the object in a semantic map, as opposed to the object pattern itself which refers to a given model learned from a classifier trained in a set of ‘local’ image descriptors. In short, the contextual information, incorporated into the pedestrian detection system, is based on prior knowledge of the object position in a semantic map of the roads.

In the decision module, shown in Figure 1 as part of the image-based module, the contextual information is processed in the form of prior probabilities according to a MAP

Algorithm 2 SVM-Cascade training process

Input: $\mathcal{D} = \{X_P, X_N\}$, np , nn : training set, number of positives, and negatives;
Output: $\{W\}$: set of parameters of the cascade;
1: TP : true positive rate; Thr_{tp} : adopted threshold for TP e.g., $Thr_{tp} = 0.95$;
2: n_f step: increment on the number of features (in our case, $n_f = 44$);
3: HOG-features are indexed, as function of the bin, in the array S ; where $|S| = n_{fea}$;
4: $\Omega \leftarrow \{\emptyset\}$: set of true negatives;
5: $i \leftarrow 1$: cascade stage;
6: $\mathcal{D}_i \leftarrow \{X_P, X_{N-\Omega}\}$;
7: **for** $n = n_f; n < n_{fea}; n+n_f$ **do**
8: $\mathcal{D}_i^{S_i}$: training set with n features, where $S_i|i = 1, \dots, n$ is the set of features in the i^{th} cascade stage;
9: Train a linear SVM, using $\mathcal{D}_i^{S_i}$, in order to obtain the SVM parameters $W_i = (w_i, b_i)$;
10: Calculate the true positive rate TP ;
11: **while** $TP < Thr_{tp}$ **do**
12: Decrease the threshold b_i in order to increase TP ;
13: Recalculate TP using the current SVM bias b_i ;
14: **end while**
15: Classify $\mathcal{D}_i^{S_i}$, using the cascade with i stages, to obtain the set of true negatives Ω ;
16: $i \leftarrow i + 1$;
17: Detach Ω from X_N , such that $\mathcal{D}_i = \{X_P, X_{N-\Omega}\}$;
18: **end for**

decision framework. As consequence, the response of the image-based detectors, used in the decision module, has to be posed in probabilistic terms as well; more specifically, the response of the classifiers is modeled by probabilistic distributions in order to obtain class-conditional probabilities which are used in the MAP decision rule.

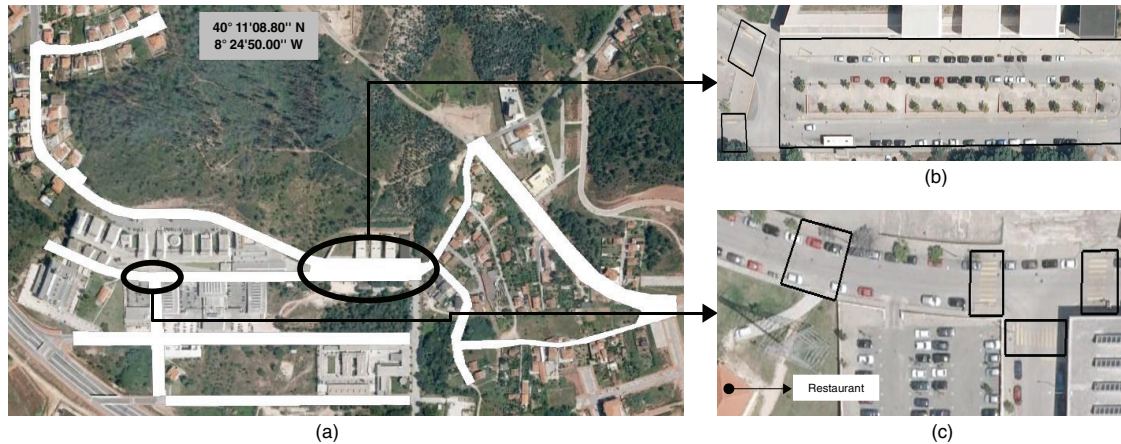


Fig. 6. Digital map of the environment where the dataset was collected. (a) Satellite image of a scenario with the map of the roads marked in white. Examples of incidence zone are given in (b) and (c).

4.1. The context-based module using object position in a semantic map

Context-based prior probabilities are used as function of the position, actual presence, of the objects in specific regions in the scenario. A semantic map of the roads traveled by the ISRobotCar was built with the aid of satellite imagery from Google Earth[®], as illustrated in Figure 6(a). A set of regions on the map was selected and identified as regions with a high potential of pedestrian occurrence. The idea was to assign a confidence score, characterized by prior knowledge, to the zones on the map that are more likely to contain pedestrians, hereafter called incidence regions. The incidence regions defined on the map are: crosswalks, regions nearby restaurants, bus stops, and cafeterias, and the zone which covers the roundabout in front of the main secretariat building. Some examples of incidence zones are shown in Figure 6(b) and Figure 6(c).

The map of the roads, and consequently the incidence regions, are defined in GPS coordinates, more specifically, GPS data is defined in terms of latitude, longitude and altitude (*lla*) in the World Geodetic System 1984 (WGS84). However, the objects (segments) detected in the LIDAR-based module have coordinates defined by a local Cartesian system. Thus, it is necessary to establish a correspondence between the GPS coordinate system and the 'local' coordinate system of the laser scanner, and vice versa.

A solution for obtaining the object's coordinates, in the LIDAR reference system, with respect to a point in the map is to convert the GPS coordinates into local navigation coordinates. This 'local' navigation system is determined by the ENU reference system. Therefore, it is necessary to transform the regions of the map and the object's coordinates to a common ENU reference system in order to determine if a detected object, in a given frame, is inside an incidence region or not.

Using the Google Earth[®] software, the regions on the map were manually drawn as polygons in satellite images

of the scenario. Thereafter, the position of these regions, defined by four pairs of points in GPS coordinates, were converted to the ENU reference system. Regarding the position of a given detected object, defined in Cartesian coordinates centered in the laser, it has to be converted to a reference point in the vehicle, aligned with the GPS rover station, and then converted to the same ENU reference system used on the map. In summary, the transformation of any point P_{GPS} in GPS coordinates² to ENU is a three-stage process:

1. Define a reference point in the map, here denoted by P_{ref} , in *lla* coordinates;
2. Express P_{GPS} , defined in *lla* coordinates, in Earth Centered Earth Fixed (ECEF) coordinates. In this work, the *lla2ecef* Matlab[®] function was used for this purpose; in Matlab[®] notation: $P_{ECEF} = \text{lla2ecef}(P_{GPS}, 'WGS84')$;
3. Convert P_{ECEF} to ENU coordinates. Denoting by P_{ENU} a point in the navigation coordinates, in Matlab[®] notation we have:

$P_{ENU} = \text{ecef2lv}(P_{ECEF}, P_{ref}, \text{ellipsoid})$, where *ellipsoid* represents the ellipsoid fitted around the Earth's globe. Using the Matlab[®] *almanac* function: *ellipsoid* = *almanac* ('earth', 'ellipsoid', 'kilometers', 'WGS84').

A differential GPS (DGPS), which operates with a ground-referenced station (base station), has an absolute precision measurement that is much more accurate than a GPS system. The base station communicates with the moving station (mounted on the vehicle) at UHF. The information shared between the stations has an average cycle of 200 ms (5 Hz), which supports the Real Time Kinematic (RTK) designation. However, DGPS is not immune to errors. In our database, some position measurements suffered from the multipath, or occlusions, problem. Moreover, a lack of reliable DGPS signal in some parts of the trajectory occurred due to temporary loss of communication between the rover and the base station. To reduce the uncertainties and errors of GPS systems, a multi-sensor fusion approach such as the one proposed by Bento

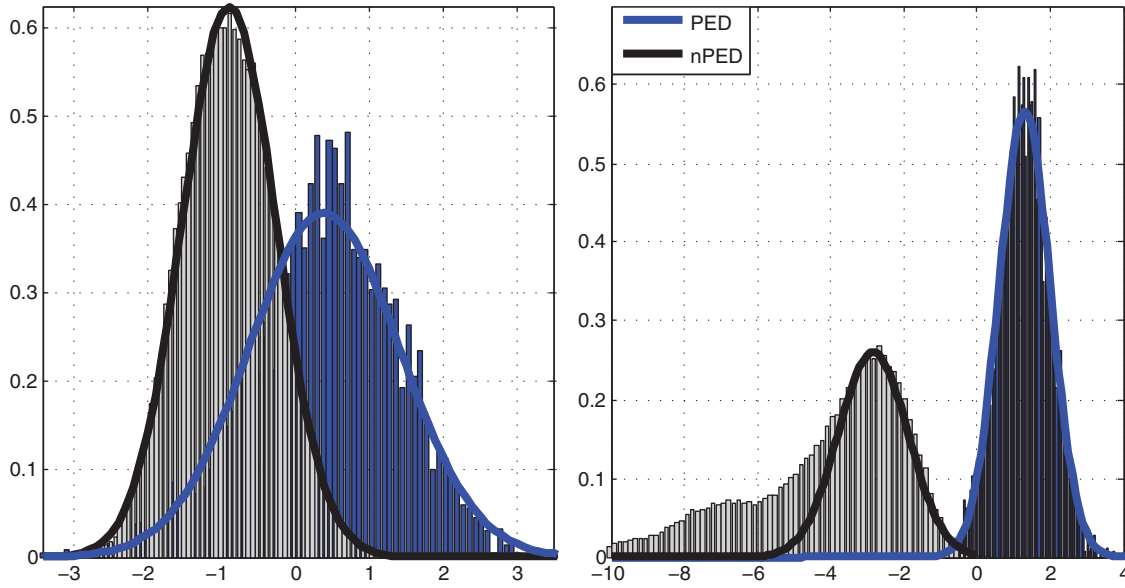


Fig. 7. Histogram and Normal distribution fitted to the output scores, on the training set, of a linear SVM (left part) and deformable parts-based model (right part).

et al. (2012), can be used to estimate and to correct the position measurements of DGPS units.

4.2. MAP decision rule using contextual information

As described in previous sections, the contextual information is processed in terms of prior probabilities. Therefore, the class-conditional probability, or simply likelihood, is obtained from the classification method used in the decision module. Thus, the outputs of the image-based detectors, obtained from the training set, were modeled according to a probabilistic distribution. Figure 7 shows the distributions of a linear SVM and the parts-based model. The histograms, plotted as function of the classifier's scores, were modeled using a Normal distribution which will be used as a likelihood function in the MAP decision. In this case, the problem resumes to the particular case of decision making using univariate–Normal densities. Designating by $P(\omega_1)$ the prior probability of a pedestrian occurrence in an incidence zone, for non-incidence zones the prior was considered to be mutually exclusive and exhaustive so, if an object is located in a non-incidence zone, its prior probability is equal to $1 - P(\omega_1)$. If $P(\omega_1) = P(\omega_0)$, the decision resumes to the case of the Maximum Likelihood (ML) rule.³ If the prior probabilities are not equal, that is, $P(\omega_1) \neq P(\omega_0)$, the decision threshold shifts away from the more likely class.

Denoting by $T_\Lambda = 0$ the decision threshold when $P(\omega_0) = P(\omega_1)$, as the value of $P(\omega_i | i=\{0,1\})$ varies during the experiments using context, T_Λ also varies. Let $\mathcal{N}(\mu_1, \sigma_1^2)$ and $\mathcal{N}(\mu_0, \sigma_0^2)$ be the Normal distributions for the positive and negative classes respectively, and denoting the posterior probability of the event being a pedestrian by

$P(\omega_1 | x \geq 0)$, the Bayes' formula is expressed by

$$P(\omega_1 | x) = \frac{p(x | \omega_1) P(\omega_1)}{p(x | \omega_1) P(\omega_1) + p(x | \omega_0) P(\omega_0)} \quad (4)$$

where $p(x | \omega_i)$, $i = \{0, 1\}$ are the class-conditional probability density functions modeled by the Normals $\mathcal{N}(\mu_i, \sigma_i^2)$. Rearranging equation (4), we have

$$p(x | \omega_1) = p(x | \omega_0) \frac{P(\omega_1 | x) (1 - P(\omega_1))}{P(\omega_1) (1 - P(\omega_1 | x))} \quad (5)$$

Using the natural logarithm in both sides of equation (5) and expanding the squares, it follows that

$$\frac{(x^2 - 2x\mu_1 + \mu_1^2)}{2\sigma_1^2} = \frac{(x^2 - 2x\mu_0 + \mu_0^2)}{2\sigma_0^2} - \ln\left(\frac{\sigma_1}{\sigma_0} \mathcal{P}\right) \quad (6)$$

where $\mathcal{P} = P(\omega_1 | x) (1 - P(\omega_1)) / P(\omega_1) (1 - P(\omega_1 | x))$. Thus, the value of the decision boundary's threshold for any $P(\omega_1) > 0$ is the solution of the quadratic equation:

$$2(\sigma_1^2 - \sigma_0^2)x^2 + 4(\sigma_0^2\mu_1 - \sigma_1^2\mu_0)x - (4\sigma_1^2\sigma_0^2W + 2\sigma_0^2\mu_1^2 - 2\sigma_1^2\mu_0^2) = 0 \quad (7)$$

where

$$W = \ln \frac{\sigma_1(P(\omega_1 | x = 0) - P(\omega_1)P(\omega_1 | x = 0))}{\sigma_0(P(\omega_1) - P(\omega_1)P(\omega_1 | x = 0))} \quad (8)$$

Equation (8) is valid when the detected object is inside an incidence zone, otherwise $P(\omega_1)$ should be replaced by $P(\omega_0)$. Notice that the variable x denotes the classification score, where $x = 0$ is the 'optimal' threshold when $p(x | \omega_1) = p(x | \omega_0)$. As the priors are changed, the thresholds change as well. If $P(\omega_1) > P(\omega_0)$ the decision boundary shifts away from the more likely class, ω_1 , and vice versa.



Fig. 8. On the right, the ISRobotCar vehicle. On the left, an example of a set of ROIs marked as dashed boxes.

5. Experiments

The main objective of the experiments reported in this paper is to evaluate the detection performance of the system regarding (i) the classification method used in the image-based module, and (ii) as function of the prior probabilities that characterize the incidence zones in the semantic map. To support experimental analysis, the multisensor LIPD dataset was used (see Section 5.2 for details). In Section 5.3, the classification methods are evaluated without contextual information. The influence of priors on the MAP decision rule is shown using the experimental results presented in Section 5.4. To ensure coherent results among the evaluated methods, all the parameters and variables of the pre-processing and segmentation stages used in the LIDAR-based module were rigorously the same during the experiments. In other words: the results reported in the sequel depend, exclusively, on the classification methods and on the context-based approach.

5.1. Criterion used for performance evaluation

Per-frame evaluation was chosen as the preferred methodology for evaluation of the pedestrian detection system, as used by Dollar et al. (2009), Enzweiler and Gavrila (2009) and Enzweiler and Gavrila (2011). Therefore, the basis for the system performance evaluation relies on comparing all the detection windows in a given frame with the set of ground-truth $\mathcal{G}$ in the same frame. An element of $\mathcal{G}$ is defined by an area $\mathcal{A}_j^{\mathcal{G}}$ and an associated label: class-0 (occluded or under a minimum scale) or class-1 (entire body pedestrian). The decision process necessary to establish the correspondence between the detection windows and the ground-truth uses the following ratio:

$$\Upsilon(i, j) = \frac{\text{Area}(\mathcal{A}_j^{\mathcal{G}} \cap \mathcal{A}_i)}{\text{Area}(\mathcal{A}_i \cup \mathcal{A}_j^{\mathcal{G}})} > A_{\text{thr}} \quad (9)$$

i.e. the area $\mathcal{A}_i$ of a given detector has a match with a ground-truth $\mathcal{A}_j^{\mathcal{G}}$ if $\Upsilon(i, j)$ exceeds the threshold A_{thr} , where $A_{\text{thr}} = 0.25$ was chosen based on the results reported by Enzweiler and Gavrila (2009). Each detection event in $\mathcal{DW}$ should be matched at most once with an element of $\mathcal{G}$. Unmatched elements of $\mathcal{DW}$ count as false positives (*FP*), and unmatched elements of $\mathcal{G}$, labeled as class-1, count as missings. On the other hand, elements of $\mathcal{G}$ labeled as class-0 are ignored, that is, these bounding boxes do not need to be matched; however, when a match occurs it is not counted as *FP*.

5.2. The LIPD dataset

The LIPD dataset, designated $\mathcal{D}_{\text{LIPD}}$, comprises two sets, $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{test}}$; the former is used for training purposes, and the latter constitutes the testing set. The entire set⁵ contains, besides monocular images and LIDAR scans, data from two proprioceptive sensors, an Inertial Measurement Unit (IMU) and an incremental encoder, and also data from a DGPS. The dataset was recorded using the sensor acquisition system mounted on the ISRobotCar, an instrument Yamaha vehicle shown in Figure 8, driving through the areas of the Engineering Campus of the University of Coimbra and in the neighborhood (see <http://www.isr.uc.pt/~cpremeida/PoloII-Google-map.pdf>).

Due to the fact that the dataset was obtained in outdoor conditions, and since the sensor apparatus has been exposed to weather and environmental conditions, not unexpectedly, some ‘difficulties’ have occurred, such as: light exposure variations, vibrations, oscillations, noise, dust and particles in the air, among others. Perhaps one of the main problems during the data recording was the occurrence of some spots in the images due to dust on the lens.



Fig. 9. LIPD dataset examples: (first row) labeled pedestrians used in the training set; (middle row) full images in the testing set; (last row) screen shot of ROI projections, blue rectangles, in testing images.

The training part of the dataset contains 4606 manually labeled positives (image cutouts of pedestrians, upright, full body), and 2444 full-frame 640×480 resolution images without any pedestrian evidence. Thus, the elements of the training set are the aforementioned 4606 positive cutouts (or bounding boxes) and a free number of negative instances which can be extracted from the negative frames. The current testing set contains 4823 full-frame images, and detailed annotations regarding the pedestrian's appearance (in terms of occlusion), namely: occluded/partial pedestrians (class-0) and entire body pedestrians (class-1). A summary of the dataset is given in Table 1.

5.3. Performance evaluation of the image-based detectors

A linear SVM, the proposed SVM-cascade and the deformable parts-based model (hereafter called DPM), were evaluated and their detection performance on the LIPD-testing set ($\mathcal{D}_{\text{test}}$) is reported in the sequel. The classifier's models were learned using all the positive examples and a subset of 'hard' negatives extracted from $\mathcal{D}_{\text{train}}$ using the proposed SVM-based down-sampling algorithm presented in Section 3.1. Detection rate versus false positives per frame was the condition used to evaluate the performance of these methods in $\mathcal{D}_{\text{test}}$. In order to analyze, in particular, the generalization capability of the DPM method and also to have a fair comparison with the other methods, this detector was evaluated using a model learned on the LIPD-training set (designated by

Table 1. The statistics of the $\mathcal{D}_{\text{LIPD}}$ set.

Training set			
Set	N_{pos}	N_{neg}	Description
$\mathcal{D}_{\text{train}}$	4606	2444	Sunny days, autumn season. Negative instances should be extracted from the 2444 frames.
Testing set			
$\mathcal{D}_{\text{test}}$	698	*	Sunny days, autumn season. The number of negatives (*) depends on the detection approach to be used (which can take advantage of the LIDAR information or not).

DPM-1) and the model originally learned on the INRIA dataset available at <http://www.cs.brown.edu/~pff/latent-release4/>, which we called DPM-2. Both models are shown in Figure 10.

To evaluate detection performance, with no use of context, on entire images in contrast with ROI-based images, we consider the DPM method and the codes available at <http://www.cs.brown.edu/~pff/latent-release4/> as a benchmark. Moreover, to demonstrate the impact of the ratio of intersection area and union area, see equation (9),

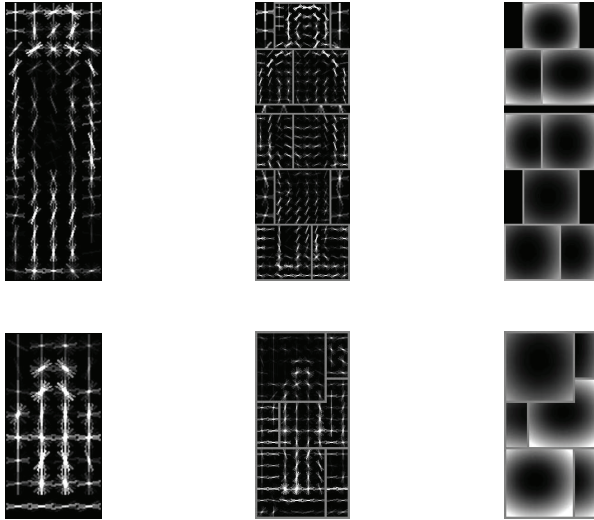


Fig. 10. Pedestrian models obtained with the DPM method. First row represents the INRIA-based model (DPM-2), which is available at <http://www.cs.brown.edu/~pff/latent-release4/>. The second row shows the model obtained with the LIPD dataset (DPM-1). From left to right: the root filter's model, followed by part filters, and spatial model for the location of each part relative to the root.

on the performance assessment, we have conducted experiments for two values of A_{thr} , 0.25 and 0.15. The results, for both models, are given in Figure 11. The performance on ROI-based images, shown in black, is better than the full-image case, gray dashed lines, for mainly two reasons: the scales are limited, i.e. the maximum size of the detection window is restricted to the size of the ROI, and the number of false positives tends to be smaller in ROI-based approaches. Regarding the models, the results were favorable to the LIPD-based model (DPM-1). One reason is due to the size of labeled pedestrians in the INRIA database, where the positives are defined by bounding boxes of 64×128 pixels, which is more than two orders of magnitude larger than the minimum positive bounding boxes of the LIPD dataset (27×54).

5.4. Performance evaluation using contextual information

In the experiments using information from the position of the objects in the semantic map, the prior probability for the positive class $P(\omega_1)$, when the object lies in an incidence zone, was varied from 0.6 to 0.9 in intervals of 0.1. Conversely, $P(\omega_0) = 1 - P(\omega_1)$ assuming mutually exclusive events. Notice that when $P(\omega_1) = P(\omega_0)$ both events are equiprobable, i.e. the contextual information has no effect on the MAP rule and the detection problem resumes to one of ML decision making. We have set $P(\omega_1) = P(\omega_0) = 0.5$ for this case.

The impact of contextual information on detecting pedestrians in ROIs is shown in Figure 12, where a linear SVM

(linSVM), the proposed SVM-cascade ensemble (SVM-casc) and the DPM method trained on the LIPD training set (DPM-1) were evaluated. The best performance, using context, was achieved with $P(\omega_1) = 0.8$, denoted by $P=0.8$ on the legend. On the other hand, results when context does not influence the decision making i.e. for $P=0.5$, are plotted on the same figure to enable comparison. In addition, we observed from additional experiments that for $P(\omega_1) > 0.8$ the detection performance tends to decrease, indicating a risk of providing the system with very confident priors. The context-based solution obtained, on average, an improvement of above 5% on the detection performance of all classification methods at the interval of 1 to 10 *FP* per frame. For some values of *FP* in that interval, a gain of over 10% in the detection rate was achieved. However, the increment on the performance was not strictly the same in all cases because the distributions that model the classifier's response are, as expected, not the same.

5.5. Discussion of processing time

To assess the processing effort, or CPU time, between pedestrian detection on full-image frames versus ROI-based approaches, we performed experiments using a linear SVM and the Fisher's Linear Discriminant classifiers. Experiments with full frames demanded, on average, a computational time three times higher than the ROI-based approach. However the CPU time of all the LIDAR-based processing stages involved in our system was not strictly taken into account in our experiments.

We also did runtime analysis between the SVM-cascade and a linear SVM, which was of the order of five times faster than the latter method. Nevertheless, both methods are similar in terms of classification performance as shown in Figure 12. On the other hand, the majority of the computational cost imposed by the context-based module depends on the processing time involved in the LIDAR-based module.

6. Conclusions

This work presented a context-based multisensor fusion architecture for pedestrian detection in urban environments. The proposed architecture comprises three main processing stages: (i) a LIDAR-based module acting as the primary object detector and ROI generator, (ii) an image-based module using sliding window detectors with the role of validating the presence of pedestrians in ROIs, and (iii) a module which supplies the system with contextual information obtained from a semantic map. The LIPD dataset, which is highly imbalanced, was used for experimental evaluation. Moreover, a down-sampling method, using support vectors extracted from multiple linear SVMs, was used to reduce the cardinality of the training set and, as a consequence, to decrease the CPU time during the training process.

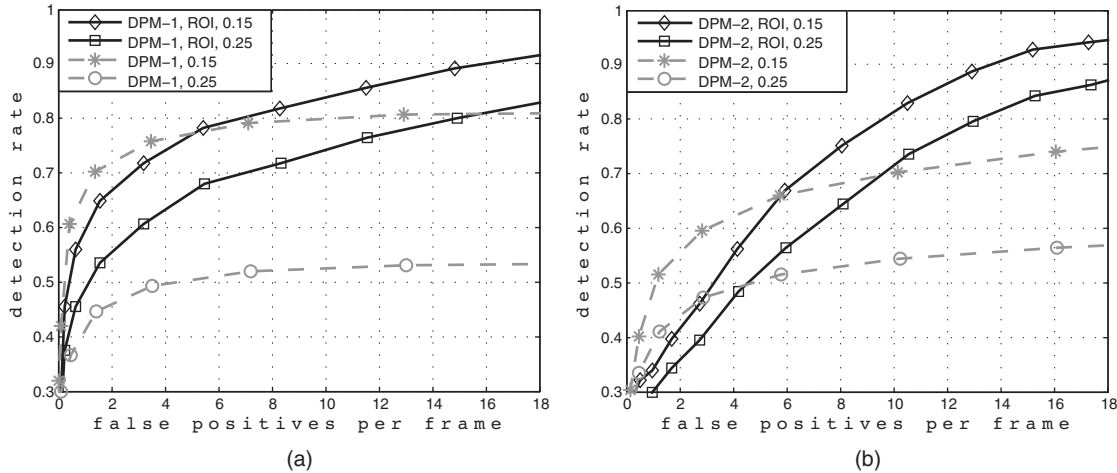


Fig. 11. Evaluation of the DPM benchmark method on ROIs obtained from the LIDAR-based module (curves in black), and on full frames (dashed curves). In (a) we have detection performance using the DPM-1 model, and in (b) the results were obtained with DPM-2. The numbers 0.25 and 0.15, on the legend, correspond to values of A_{thr} for ground-truth matching.

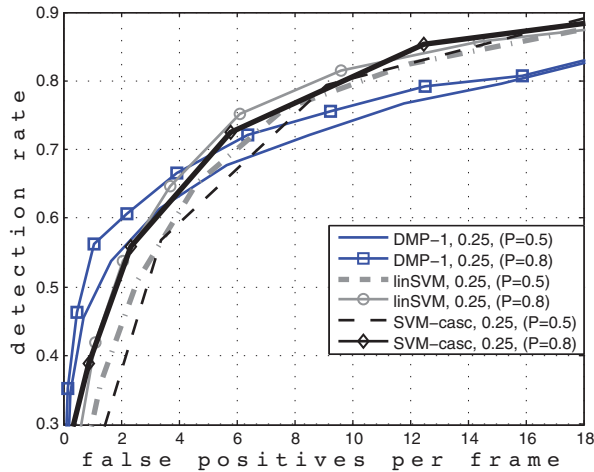


Fig. 12. Evaluation of the pedestrian detectors, in ROIs, using context-based information (indicated by $P = 0.8$) and when context does not affect the MAP decision (given by $P = 0.5$).

Experiments demonstrated that contextual information enhances the performance of a pedestrian detection system. The experiments were performed using a linear SVM, the proposed SVM-cascade and the DPM method. However, similar tendencies in the performance are expected when using other classification methods since the contextual information enters in the system as prior probabilities which are independent of the conditional probabilities given by the classifier. The use of incidence regions clearly enhanced the detection performance, which demonstrates that the probability of pedestrian occurrence in some specific zones (e.g. crosswalks, bus stops) is higher than in other zones. The use of context in such perception systems seems to be a promising area, since plenty of contextual information, other than the critical regions, can be

used, such as: day, time, weather conditions, object velocity, infrastructure-based information, and so on.

Notes

1. HOG denotes Histogram of Oriented Gradients, introduced by Dalal and Triggs (2005).
2. the point may represent the position of a detected object or one of the corners of a polygon that defines an incidence region.
3. Both classes are likely to occur.
5. available online: <http://www.isr.uc.pt/~cpremebida/dataset>

Acknowledgements

The authors thank the reviewers for their insightful comments and valuable suggestions.

Funding

This work was supported by Fundação para a Ciência e a Tecnologia (FCT) and COMPETE Program (grants PTDC/SEN-TRA/099413/2008 and PTDC/EEA-AUT/113818/2009).

References

- Bento LC, Parafita R and Nunes U (2012) Inter-vehicle sensor fusion for accurate vehicle localization supported by V2V and V2I communications. In: *15th IEEE international conference on intelligent transportation systems (ITSC '12)*, Anchorage, USA, 16–19 September 2012, pp. 907–914. Piscataway: IEEE Press.
- Dalal N and Triggs B (2005) Histograms of oriented gradients for human detection. In: *IEEE conference on computer vision and pattern recognition (CVPR '05)*, Washington DC, USA, pp. 886–893. San Diego, CA, USA. Piscataway: IEEE Press.
- Dollar P, Wojek C, Schiele B and Perona P (2009) Pedestrian detection: A benchmark. In: *IEEE conference on computer vision and pattern recognition (CVPR '09)*, Los Alamitos, USA, pp. 304–311. Miami Beach, Florida, USA. June 20–25, 2009. Piscataway: IEEE Press.

- Dollar P, Wojek C, Schiele B and Perona P (2012) Pedestrian detection: An evaluation of the state of the art. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34(4): 743–761.
- Douillard B, Fox D, Ramos F and Durrant-Whyte H (2011) Classification and semantic mapping of urban environments. *The International Journal of Robotics Research* 30(1): 5–32.
- Drake S (2002) Converting GPS coordinates (lla) to navigation coordinates (ENU). Technical Report, DSTO-TN-0432, DSTO Electronics and Surveillance Research Laboratory, Australia.
- Duda RO, Hart PE and Stork DG (2001) *Pattern Classification*. New York: John Wiley & Sons.
- Enzweiler M and Gavrilu D (2011) A multilevel mixture-of-experts framework for pedestrian classification. *IEEE Transactions on Image Processing* 20(10): 2967–2979.
- Enzweiler M and Gavrilu DM (2009) Monocular pedestrian detection: Survey and experiments. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 31(12): 2179–2195.
- Faouzi NEE, Leung H and Kurian A (2011) Data fusion in intelligent transportation systems: Progress and challenges - a survey. *Information Fusion* 12(1): 4–10.
- Felzenszwalb PF, Girshick RB, McAllester D and Ramanan D (2010) Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32: 1627–1645.
- Gandhi T and Trivedi M (2007) Pedestrian protection systems: issues, survey, and challenges. *IEEE Transactions on Intelligent Transportation Systems* 8(3): 413–430.
- Garcia F, de la Escalera A, Armingol J, Herrero J and Llinas J (2011) Fusion based safety application for pedestrian detection with danger estimation. In: *14th international conference on information fusion (FUSION '11)*, pp. 1–8. Chicago, Illinois, USA. 05–08 July, 2011. IEEE.
- Geronimo D, Lopez AM, Sappa AD and Graf T (2010) Survey on pedestrian detection for advanced driver assistance systems. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32(7): 1239–1258.
- Graf HP, Cosatto E, Bottou L, Durdanovic I and Vapnik V (2005) Parallel support vector machines: The cascade SVM. In: *Advances in Neural Information Processing Systems* (ed Y Weiss, B Schölkopf and J Platt), vol. 18, pp. 521–528. Vancouver and Whistler, British Columbia, Canada. December 13–18, 2004. Cambridge, MIT Press.
- Heikkilä J and Silven O (1997) A four-step camera calibration procedure with implicit image correction. In: *IEEE conference on computer vision and pattern recognition (CVPR '97)*, pp. 1106–1112. San Juan, Puerto Rico. June 17–19, 1997. Piscataway: IEEE Press.
- Kang P and Cho S (2006) EUS SVMS: Ensemble of under-sampled SVMS for data imbalance problems. In: Irwin King, Jun Wang, Lai-Wan Chan, DeLiang Wang. *Neural Information Processing*. Berlin: Springer, vol. 4232, pp. 837–846.
- Ludwig O, Premebida C, Nunes U and Araujo R (2011) Evaluation of boosting-SVM and SRM-SVM cascade classifiers in laser and vision-based pedestrian detection. In: *IEEE international conference on intelligent transportation systems (ITSC '11)*, pp. 1574–1579. October, 05–07, 2011. Washington, DC, USA. Piscataway: IEEE Press.
- Markoff J (2010) Google cars drive themselves, in traffic. *New York Times*, October 9, 2010.
- Navarro-Serment LE, Mertz C and Hebert M (2010) Pedestrian detection and tracking using three-dimensional LADAR data. *The International Journal of Robotics Research* 29(12): 1516–1528.
- Perko R and Leonardis A (2010) A framework for visual-context-aware object detection in still images. *Computer Vision and Image Understanding* 114(6): 700–711.
- Silva M, Moita F, Nunes U, Garrote L, Faria H and Ruivo J (2012) ISRobotCar: The autonomous electric vehicle project. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 7–12 October 2012: 4233–4234.
- Spinello L, Triebel R and Siegwart R (2010) Multiclass multimodal detection and tracking in urban environments. *The International Journal of Robotics Research* 29(12): 1498–1515.
- Stiller C, León FP and Kruse M (2011) Information fusion for automotive applications - an overview. *Information Fusion* 12(4): 244–252.
- Vasconcelos F, Barreto J and Nunes U (2012) A minimal solution for the extrinsic calibration of a camera and a laser-rangefinder. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 34(11): 2097–2107.
- Viola P and Jones M (2001) Rapid object detection using a boosted cascade of simple features. In: *IEEE conference on computer vision and pattern recognition (CVPR '01)*, Hawaii, USA, vol. 1, pp. 511–518. 8–14 December 2001. Piscataway: IEEE Press.