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

Cristiano Premebida and Urbano Nunes

This is a pre-print version.
The final version is available at: http://ijr.sagepub.com/content/32/3.toc

Abstract

In this work, a context-based multisensor system, applied for pedestrian detection in urban environment, is presented. The proposed system comprises three main processing modules: (i) a LIDAR-based module acting as primary object detection, (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 regions of interest (ROIs) generated by the LIDAR module. A Bayesian strategy is used to combine information from sensors on-board the vehicle ('local' information) with information contained in a digital map of the roads ('global' information). To support experimental analysis, a multisensor dataset, named Laser and Image Pedestrian Detection dataset (LIPD), is used. The LIPD dataset was collected in an urban environment, at day light 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 consequence, to decrease the CPU-time during the training process of image-based classifiers. The performance of the system is evaluated, in terms of true positive rate and 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.

1 Introduction

Multisensor data fusion plays an important role in the field of Intelligent Transportation Systems (ITS) and Intelligent Vehicles (IV), evidenced by a significant number of work in these areas, for instance the recent surveys of Faouzi et al. (2011), Stiller et al. (2011), Dollar et al. (2012) and Geronimo et al. (2010). Advanced Driver Assistance Systems (ADAS) have deserved much attention in the recent years because of many applications in the automotive market *e.g.*, Adaptive Cruise Control (ACC), Lane Departure Warning (LDW), Anti-lock Braking

System (ABS), Collision Warning Systems (CWS), and Pedestrian Protection Systems (PPS). The later, which is emphasized in this work, can be divided, in general words, in 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 primary to mitigate possible injuries on pedestrians due to an impact *e.g.*, special designed front bumper, deformable hood, specific air-bags placed nearby the frontal columns of the vehicle, and so on. Whereas, active pedestrian protection systems are based on sensors on-board the vehicle, and/or on the infrastructure, with the role of predicting and anticipating possible risks of collisions.

On-board sensor-based pedestrian detection systems, see the relevant survey of Gandhi and Trivedi (2007), in the domain of ADAS applications, is a research topic which has received considerable attention, evidenced by the works of Markoff (2010), Navarro-Serment et al. (2010), Spinello et al. (2010), Douillard et al. (2011), Felzenszwalb et al. (2010), 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 among objects as proposed by Perko and Leonardis (2010), or contextual information like scale, distance, and road location, which can be obtained by LIDARs or stereo-vision as discussed by Geronimo et al. (2010). The aforementioned works share a common element: the information is extracted from on-board sensors. On the other hand, solutions for pedestrian detection combining 'external' sources of contextual information *e.g.*, GPS-based semantic map, seems to be an interesting approach that have been rarely addressed, if ever, 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 on-board an electrical vehicle (see Fig. 8), with information obtained from a semantic map of the roads driven by the vehicle. 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 stop. The multisensor information is processed using an architecture which comprises three main processing modules: (i) LIDAR-based module; (ii) Context-based module; and (iii) Image-based module. The LIDAR-based module is in charge of primary objection 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 position of the objects in the map is used as contextual information. Since the laser is calibrated *w.r.t.* the camera, the position of the set of objects are used to project a set of regions of interest (ROIs) in the image plane. Inside each ROI, a image-based classifier is

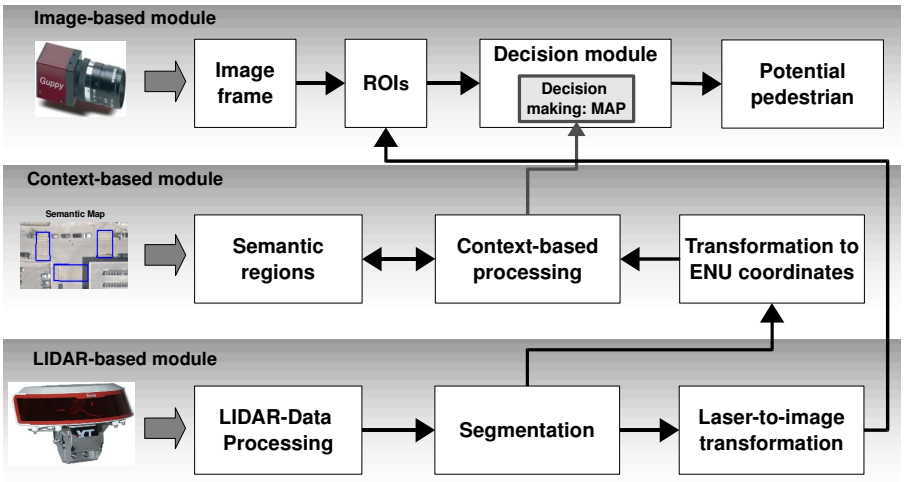


Figure 1: Block diagram illustrating the main processing modules comprised in the pedestrian detection system with contextual information incorporated into the system.

used in the form of a multiscale sliding-window detectors which are shifted in position and size for searching 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-SVM and with the Deformable Parts-based Models proposed by Felzenszwalb et al. (2010) and available on Felzenszwalb et al. (2012). 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 Fig. 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 LIPD dataset, is reported in Section 5. Finally, Section 6 presents the conclusions.

2 LIDAR-based module

Our LIDAR based system acts as 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 Fig. 2. Concisely, the main processing modules performed in the LIDAR-based module are:

1. Pre-segmentation and filtering: comprehends a set of pertinent data processing tasks, necessary to decrease complexity and processing time of subsequent stages, such as: filtering-out isolated/spurious range-points, discarding measurements that occur out a predefined 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: the 2D Cartesian dimensions of the segments are transformed to a 'local' navigation system, 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 evolving of the laser data thought 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 laserscanner manufacture.

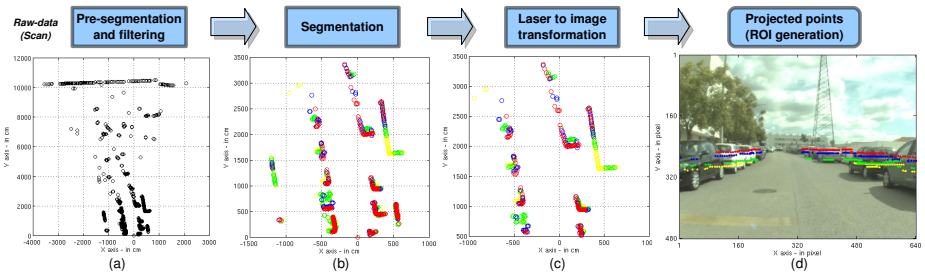


Figure 2: Main processing steps in the LIDAR-based module. (a) Raw range-points in Cartesian coordinates. (b) Points are filtered and grouped per layer, where each color indicates a given layer. (c) Range-points outside the camera FOV are discarded. (d) Projection of the points, per layer, on the image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

The laser and the monocular camera were calibrated according to the following steps: (1) a calibration dataset, with synchronized laser scans and image frames, were collect. The platform was held stationary, while the checkerboard was positioned from 2 to 7 meters away the platform; (2) the intrinsic and extrinsic parameters of the camera were estimated using the Bouguet (2007)

calibration toolbox; (3) finally, the extrinsic parameters of the camera *w.r.t.* the laser was 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 3x3 orthonormal rotation matrix representing the camera's orientation relative to the laser and T_C^L is the 3- 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 Heikkila and Silven (1997). Finally, the pixel coordinates in the image plane is 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:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K \begin{bmatrix} X_n(1) \\ X_n(2) \\ 1 \end{bmatrix} \tag{1}$$

with the camera matrix K given by

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

where fc 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 $2.5m$ as the maximum height of a pedestrian. The following matrix, necessary to make a rigid correspondence between the laserscanner and the camera reference system, was obtained:

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

where the translational vector components are in mm.

3 Image-based module

The pedestrian detection system involves a number of spatio-temporal processing techniques aiming 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 amount of negatives (potential false alarms) in contrast with the number of positives, what demands vast processing time and a high confidence detector. The adopted approach 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 pedestrian evidence. This approach decreases the computational processing time, restricting the zones of interest on the image to a dozen of ROIs at most, and reducing the false positives.

The Image-based system discussed in this work is composed by a 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: a 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.

3. Decision making: a MAP decision rule, integrating the confident scores α 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 HOG[1] descriptors, whereas in the first two methods we have used the codes available by Dollar et al. (2012) to extract HOG features. A detection window, denoted by $\mathcal{DW}_i$, is used under a multiscale sliding approach thus, $\mathcal{DW}_i$ is shift inside each ROI by varying the location (x_i, y_i) and the size (w_i, h_i) as function of a spatial stride steps and scale factors; see Fig. 3. A given detector $\mathcal{DW}_i = [px_i, py_i, w_i, h_i, \alpha_i]$ 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 it.

An inevitable problem that arises in 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 Enzweiler and Gavrila (2009) and Dollar et al. (2009), was used to discard multiple-detector occurrence around close/similar locations.

[1] HOG denotes Histogram of Oriented Gradients, introduced by Dalal and Triggs (2005).

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 remaining are discarded. The subsequent, and final, processing stage involved in the Image-based module is a 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.

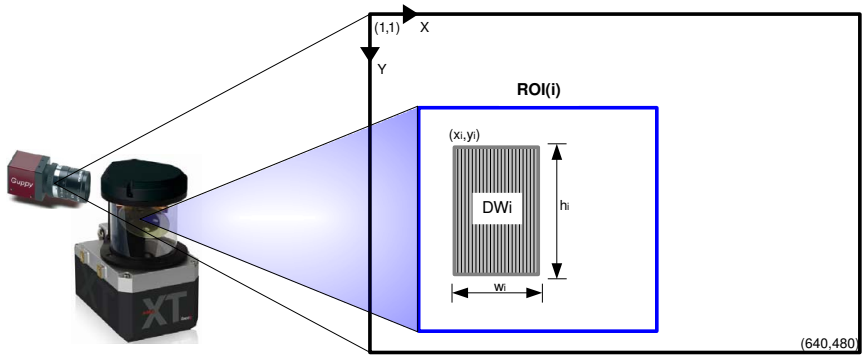


Figure 3: Representation of a detection window, $\mathcal{DW}_i$, with size defined by (w_i, h_i), which is shifted, inside a ROI, at locations given by (x_i, y_i).

It is important to punctuate 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 positives and negatives samples. To give a clear insight about this, lets consider the size (width and height in pixels) of a pedestrian detector as $\Delta_i = s_i(w, h)|\{i = 1, \cdots, 11\}$, with horizontal and vertical sliding step factors been proportional to w and h, where $h = 2w$. A pedestrian detector is shifted through the image with sizes Δ_i and at locations defined by $s_i w/4$ and $s_i h/4$. For the case of an image with 640x480 pixels, and considering $w = 27$ and the scales $s_i = (1.0, 1.2, \cdots, 2.8, 3.0)$, it generates, depending the approximations done to keep Δ with integer pixel values, at least 12K detection windows per frame. In a dataset with 10K frames, and for an average rate of 0.5 pedestrian per frame (which is a realistic factor in a typical urban scenario), it gives at least 1.2×10^8 negatives detection windows against 5K 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 to compute 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:
$X_P = \{X_{P(i)} : i \in Ip = \{1, \cdots, np\}\}$ is the input set of positive training examples.
$X_P^S = \{X_{P(i)}^S : i \in Sp \subset Ip\}$ is the subset of Selected positive training set.
$X_N = \{X_{N(i)} : i \in In = \{1, \cdots, nn\}\}$ is the input set of negative training examples.
$X_N^S = \{X_{N(i)}^S : i \in Sn \subset In\}$ is the subset of Selected negative training set.
$X_{P,N} = \{X_P, X_N\}$ 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.
$X_{N-\Omega} = \{X_{N(i)} : i \in In \backslash \Omega\}$ is the subset of negatives instances X_N with those in Ω removed, where $\Omega \subset X_N$.

The number of negative instances nn in a realistic dataset heavily outnumber 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, where $|\mathcal{SV}| = ns$. The first step of the down sampling algorithm is to split X_N on n subsets $\Omega_i \subset X_N, i = 1, \cdots, n$; further, for each subset Ω_i, a SVM classifier is used to extract the support vectors which will be used to compose $\mathcal{SV}$. Thus, each i^{th}-SVM is trained with a subset comprising n_p 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 by the negative support vectors: $X_N^S \leftarrow \mathcal{SV}$ (see Algorithm 1). Figure 4 illustrates, in the row (a), some negative samples whose training feature vectors do not correspond to support vectors and, in the row (b), some samples which correspond to $\mathcal{SV}$.

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 classifiers were trained with (i) all examples of a training set, having 100000 negatives examples, and with (ii) a reduced subset constituted of support vectors ($\mathcal{SV}$) obtained from the former training set; thus, two models have been learned. Then, we applied the classifiers on a testing set and, in 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.

3.2 SVM-cascade method

The proposed SVM-cascade is trained using a boosting process where the number of features, in a given stage, increases *wrt* to the preceding stage; thus, the complexity of the cascade and its classification capability increase as more stages are added to the structure. The SVM-cascade is a cascade of linear-

Algorithm 1 SVM-based down sampling algorithm for negative examples selection

Input: $X_{P,N} = \{X_P, X_N\}$: training set;
Output: X_N^S: set of selected negative samples;
$\{\mathcal{SV}\}_i$: set of support vectors; n: number of iterations;
Ω_i: subset of X_N;
// `Initialization:`
$X_N^S \leftarrow \{\}$: empty set;
Decompose X_N on n subsets Ω_i, where $\Omega_i \subset X_N$;
// `Selecting samples:`
for $i = 1; i < n; i{+}1$ **do**
  Train a linear-SVM classifier using $\{X_P, \Omega_i\}$;
  Use the 'negative' support vectors, from Ω_i, to generate $\{\mathcal{SV}\}_i$;
end for
// `Composing` X_N^S`:`
$X_N^S \leftarrow \{\mathcal{SV}\}_i, i = 1, \cdots, n.$

Figure 4: Negative samples which possess feature vectors positioned 'far' from the separation margin (a); and samples which correspond to support vectors (b).

SVMs in series, 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 Fig. 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. This process is performed 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 negatives 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 *wrt* the previous stage.

The feature vector used to train the SVM-cascade, and a linear-SVM, is

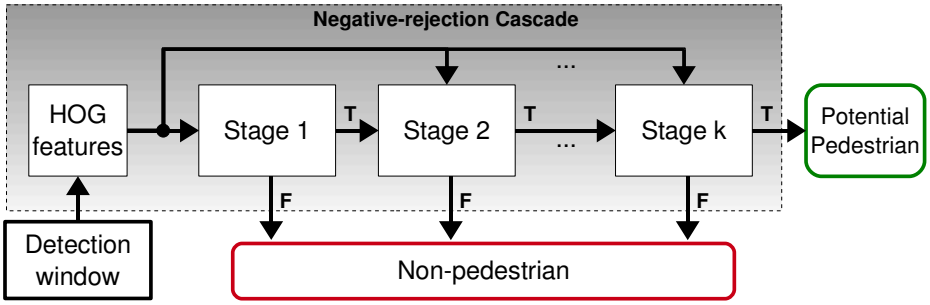


Figure 5: Functional diagram illustrating the SVM-cascade detector: each stage k rejects the samples classified as negative (F: false), while the positives (T: true) pass through all the stages to be finally detected as a pedestrian.

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;
TP: true positive rate; Thr_{tp}: adopted threshold for TP *e.g.*, $Thr_{tp} = 0.95$;
n_f step: increment on the number of features (in our case, $n_f = 44$);
HOG-features are indexed, as function of the bin, in the array S; where $|S| = nfea$;
$\Omega \leftarrow \{\emptyset\}$: set of true negatives;
$i \leftarrow 1$: cascade stage;
$\mathcal{D}_i \leftarrow \{X_P, X_{N-\Omega}\}$;
for $n = n_f; n < nfea; n+n_f$ **do**
  $\mathcal{D}_i^{S_i}$: training set with n features, where $S_i | i = 1, \cdots, n$ is the set of features in the i^{th} cascade stage;
  train a SVM, using $\mathcal{D}_i^{S_i}$, in order to obtain the SVM parameters $W_i = (w_i, b_i)$;
  calculate the true positive rate TP;
  while $TP < Thr_{tp}$ **do**
    decrease the threshold b_i in order to increase TP;
    recalculate TP using the current SVM bias b_i;
  end while
  Classify $\mathcal{D}_i$, using the cascade with i stages, and detach the true negative occurrences Ω, where $\Omega \in X_N$, to compose the training set for the next stage;
  $i \leftarrow i + 1$;
  Detach Ω from X_N such that $\mathcal{D}_i = \{X_P, X_{N-\Omega}\}$
end for

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 one bin of the grid, totalizing 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 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) book, while a description of image-based contextual

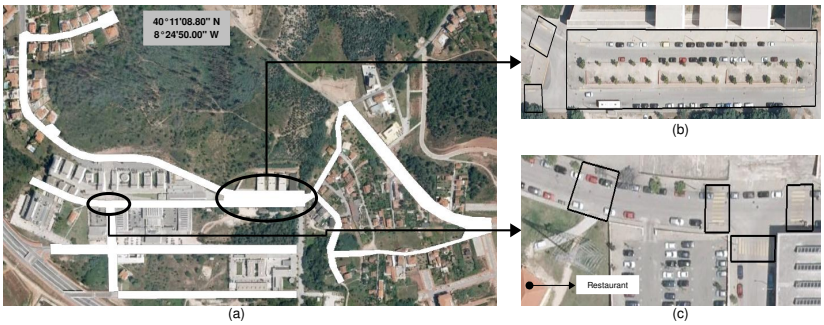


Figure 6: Digital map of the environment where the dataset were collected. (a) Satellite image of scenario with the map of the roads marked in white. Examples of incidence zones, used in the context-based module, are given in (b) and (c).

cues is given by Perko and Leonardis (2010). Here, context refers to the position 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 Fig. 1 as part of the Image-based module, the contextual information is processed in the form of prior probabilities according to a MAP 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 are modeled by probabilistic distributions in order to obtain class-conditional probabilities which are used in the MAP decision rule.

4.1 Context-based module using object position in a semantic map

Context-based prior probabilities are used as function of the position, actually 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 the Google Earth©, as illustrated in Fig. 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 a 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 Fig. 6(b) and Fig. 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 laserscanner, and *vice versa*.

A solution for obtaining the objects 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 east, north and up (ENU) reference system as described in the work of Drake (2002). Therefore, it is necessary to transform the regions of the map and the objects coordinates to a common ENU reference system in order to determine if a detected object, in a given frame, is inside or not an incidence region.

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 a ENU reference system. Regarding the position of a given detected object, defined in Cartesian coordinates centered in the laser, its 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}[2] in GPS coordinates 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} =*lla2ecef*(P_{GPS}, *'WGS84'*);
3. Convert P_{ECEF} to the ENU coordinate. Denoting by P_{ENU} a point in the navigation coordinates, in Matlab© notation we have:
P_{ENU} =*ecef2lv*(P_{ECEF},P_{ref},*ellipsoid*), where *ellipsoid* represents the Ellipsoid fitted around the Earth globe. Using the Matlab© *almanac* function: *ellipsoid* = *almanac*(*'earth'*,*'ellipsoid'*,*'kilometers'*,*'WGS84'*).

A Differential-GPS, which operates with a ground referenced station (base station), has an absolute precision measurement much more accurate than a GPS system. The base station communicates with the moving station (mounted on the vehicle) at UHF frequency. The information shared between the stations has an average cycle of 200 ms (5 Hz), which supports the RTK designation. However, DGPS is not immune to errors. In our database, some position measurements suffered with the multi-path, or occlusions, problem. Moreover, the lack of credibility of GPS in some parts of the trajectory occurred due to temporary lost of communication between the rover and the base station. To reduce the uncertainties and errors of GPS systems, a multisensor fusion approach such as the proposed by Bento et al. (2012), can be used to estimate and to correct the position measurements of DGPS units.

[2]whatever the point represents the position of a detected object or one of the corners of the polygon that defines an incidence region.

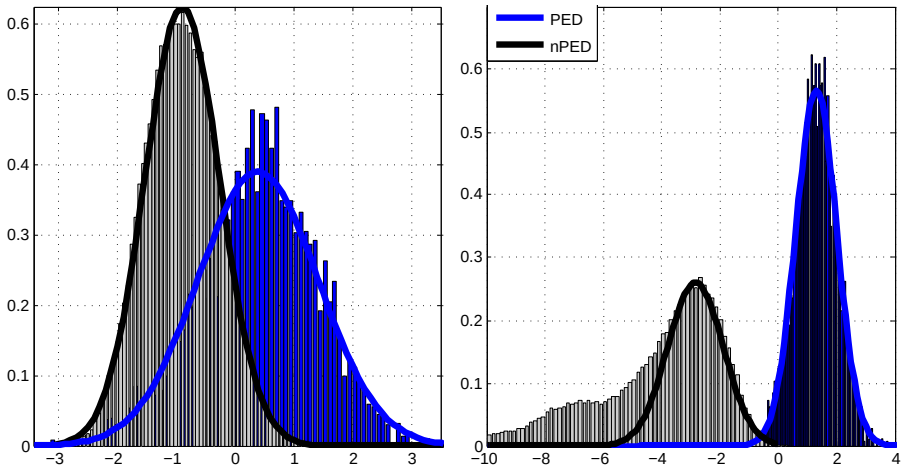


Figure 7: Histogram and Normal distribution fitted to the output scores, on the training set, of a linSVM and the Deformable Parts-based Model.

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 lin-SVM and the Parts-based Model. The histograms, plotted as function of the classifier's scores, were modeled by a Normal distribution which will be used as 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 pedestrian occurrence in an incidence zone, for the 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 is equal to $1-P(\omega_1)$. If $P(\omega_1) = P(\omega_0)$[3], the decision resumes to the case of ML rule. 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_Λ 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 be a pedestrian by $P(\omega_1|x \geq 0)$, the Bayes'formula is expressed by

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

[3]Both classes are likely to occur.

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 (4), we have

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

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

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

where $\mathcal{P} = P(\omega_1|x)(1 - P(\omega_1))/P(\omega_1)(1 - P(\omega_1|x))$, *c.f.* (5). Thus, the value of the decision boundary's threshold for any $P(\omega_1) > 0$ is the solution of the quadratic equation:

$$\begin{gathered}(\sigma_1^2 - \sigma_0^2)x^2 + 2(\sigma_0^2\mu_1 - \sigma_1^2\mu_0)x - \\ (2\sigma_1^2\sigma_0^2 W + \sigma_0^2\mu_1^2 - \sigma_1^2\mu_0^2) = 0\end{gathered} \tag{7}$$

where,

$$W = ln\frac{\sigma_1(P(\omega_1|x = T_\Lambda) - P(\omega_1)P(\omega_1|x = T_\Lambda))}{\sigma_0(P(\omega_1) - P(\omega_1)P(\omega_1|x = T_\Lambda))} \tag{8}$$

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 $x = T_\Lambda$ (usually T_Λ=0) is the 'optimal' threshold when $P(\omega_1|x) = P(\omega_0|x)$. As the priors are changed, the thresholds changed as well. If $P(\omega_1) > P(\omega_0)$ the decision boundary shifts away from the more likely class, ω_1, and *vice-versa*.

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 as function of (ii) 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. After, the influence of priors in the MAP decision rule is shown on the experimental results presented in Section 5.4. To have coherent results among the evaluated methods, all the parameters and variables of the preprocessing 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.

Figure 8: A snapshot of the ISRobotCar instrumented vehicle is depicted on the right. In the left-side, an example of a set of ROIs, marked as dashed-boxes, is shown.

5.1 Criterion used for performance evaluation

Per-frame evaluation was chosen as the preferred methodology for the 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 $\mathcal{A}_j^G$ 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 among the detection windows and the ground-truth uses the following ratio (Jaccard index):

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

i.e., the area $\mathcal{A}_i$ of a given detector has a match with a ground-truth $\mathcal{A}_j^G$ if $\Upsilon(i,j)$ exceeds the threshold A_{thr}, where $A_{thr} = 0.25$ was chosen based on the results reported by Enzweiler and Gavrila (2009); although quite arbitrary[4], this value is reasonable due to the average dimensions of the elements in $\mathcal{G}$. 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, and unmatched elements of $\mathcal{G}$, labeled as class-1, count as missing. 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.

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

5.2 The LIPD dataset

The Laser and Image Pedestrian Detection (LIPD) dataset, designated $\mathcal{D}_{LIPD}$, comprises two sets, $\mathcal{D}_{train}$ and $\mathcal{D}_{test}$; the former is used for training purposes, and the later constitutes the testing set. The entire set[5] contains, besides monocular images and LIDAR scans, data from two proprioceptive sensors, an IMU and an incremental encoder, and also data from a differential GPS. The dataset was recorded from the sensor acquisition system mounted on the ISRobotCar, an instrument Yamaha vehicle shown in Fig. 8, driving through the areas of the Technological *Campus* of the University of Coimbra and in the neighboring[6].

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 namely: light exposure variations, vibrations, oscillations, noise, dust and particles on 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.

The training part of the dataset contains 4606 manually labeled positives (image's cutouts of pedestrian in up-right entire body), and 2444 full-frames 640x480 resolution images without any pedestrian evidence. Thus, the elements of the training set are the aforementioned 5327 positives cutouts (or bounding boxes) and a free-number of negative instances which can be extracted from the negatives frames. The current testing set contains 4823 full-frame images, and

[4]Other possible value for A_{thr} is 0.5 as suggested by Dollar et al. (2009)
[5]available on the Web: `http://www.isr.uc.pt/~cpremebida/dataset`
[6]`http://www.isr.uc.pt/~cpremebida/PoloII-Google-map.pdf`

detailed annotations regarding the pedestrians 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.

Table 1: Statistics of $\mathcal{D}_{LIPD}$

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

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 the detection performance on the LIPD-testing set ($\mathcal{D}_{test}$) is reported in the sequel. The classifier's models were learned using all the positives examples and a subset of 'hard' negatives extracted from $\mathcal{D}_{train}$ 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 $\mathcal{D}_{test}$. In order to analyze, in particular, the generalization capability of the DPM method and also to have a fair comparison with the others methods, this detector was evaluated using a model learned on the LIPD-training set (designated by DPM-1) and the model originally learned on the INRIA dataset and available by Felzenszwalb et al. (2012), which we called DPM-2. Both models are shown in Fig. 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 by Felzenszwalb et al. (2012) as benchmark. Moreover, to demonstrate the impact of the ratio of intersection area and union area, see (9), 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 Fig. 11. The performance on ROI-based images, shown in black, are better than the full-image case, gray dashed-lines, due to 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 tend 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 on 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).

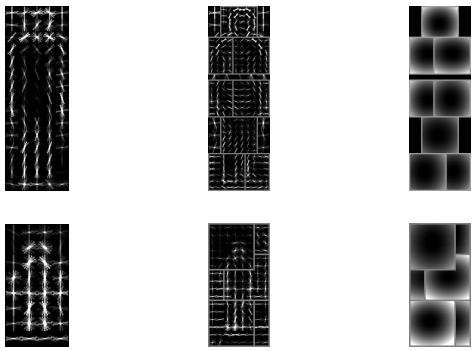
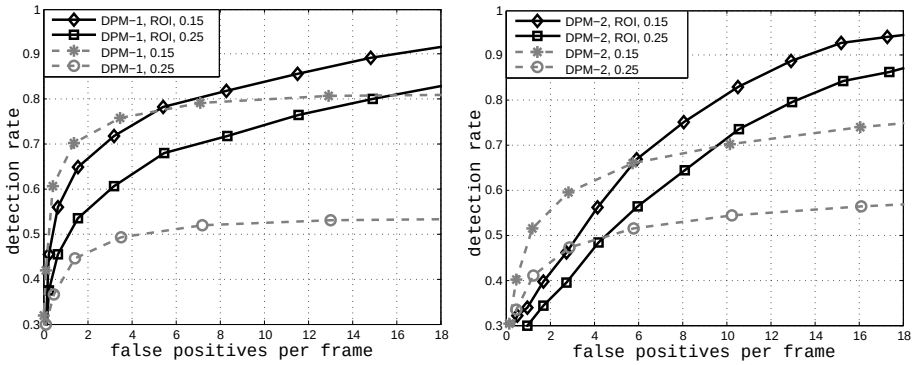


Figure 10: Pedestrian models obtained with the DPM method. First row represents the INRIA-based model (DPM-2), which is available on the website of Felzenszwalb et al. (2012). The second row shows the model obtained with the LIPD dataset (DPM-1). From left to right: the root filter's model, followed by part filters, and spatial model for the location of each part relative to the root.



(a) (b)

Figure 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 obtained with DPM-2. The numbers 0.25 and 0.15, on the legend, correspond to values of threshold A_{thr} for ground-truth matching.

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, $P(\omega_0) = 1 - P(\omega_1)$ since it was assumed the mutually exclusive condition between the 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 a ML decision making. In particular, we have set $P(\omega_1) = P(\omega_0) = 0.5$ for this case.

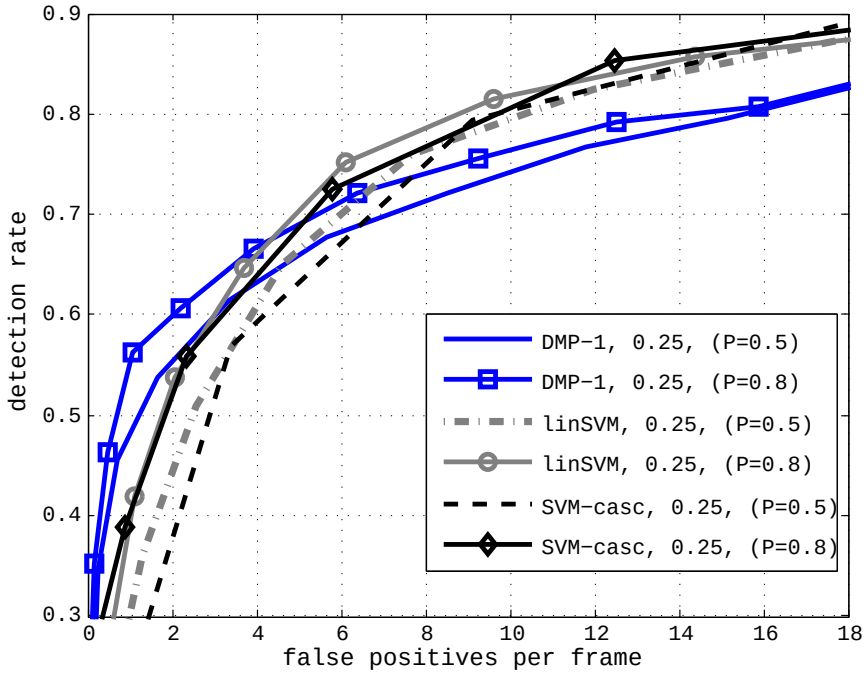


Figure 12: Evaluation of the pedestrian detectors, in ROIs, using context-based information (indicated by P=0.8) and when context does not affect the MAP decision (given by P=0.5).

The impact of contextual information on detecting pedestrians in ROIs is shown in Fig. 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 in the same figure to enable comparison. Besides, 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, in 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 classifiers 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 images frames *vs* ROI-based approaches, we performed experiments using a linear-SVM and the Fisher's Linear Discriminant classifiers. Experiments with full frames demanded, in 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 linSVM, which has been of the order of five times faster to the later method. Nevertheless, both methods are slightly similar in terms of classification performance as shown in Fig. 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 environment. The proposed architecture comprises three main processing stages: (i) a LIDAR-based module acting as primary object detection and ROI generator, (ii) a 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 high imbalanced, were 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 consequence, to decrease the CPU-time during the training process.

Experiments demonstrate that contextual information enhances the performance of a pedestrian detection system. The experiments were performed using a linear-SVM, a proposed SVM-cascade and the DPM method. However, similar tendency in the performance is expected 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 other zones. The use of context in such perception system seems to be a promising area since plenty of contextual information, other than the critical regions, can be used, such as: day-time, weather condition, object velocity, infrastructure-based information, and so on.

7 Acknowledgments

This work was supported by National Funds through Fundação para a Ciência e a Tecnologia de Portugal (FCT), under project grant PTDC/EEA-AUT/113818/2009. The authors thank the reviewers for their insightful comments and valuable suggestions.

References

Bento, L. C., Parafita, R., and Nunes, U. (2012). Inter-vehicle sensor fusion for accurate vehicle localization supported by v2v and v2i communications. In *Intelligent Transportation Systems, ITSC. 15th IEEE International Conference on*, pages 907–914, USA.

Bouguet, J. Y. (2007). Camera calibration toolbox for matlab [online]. http://www.vision.caltech.edu/bouguetj/calib_doc/.

Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, CVPR. IEEE Conference on*, pages 886–893, Washington, DC, USA. IEEE Computer Society.

Dollar, P., Wojek, C., Schiele, B., and Perona, P. (2009). Pedestrian detection: A benchmark. In *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, pages 304–311, Los Alamitos, CA, USA.

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). Dsto-tn-0432, DSTO Electronics and Surveillance Research Laboratory, Australia.

Duda., R. O., Hart, P. E., and Stork, D. G. (2001). *Pattern classification*. John Wiley & Sons, Inc, NY.

Enzweiler, M. and Gavrila, D. (2011). A multilevel mixture-of-experts framework for pedestrian classification. *Image Processing, IEEE Transactions on*, 20(10):2967–2979.

Enzweiler, M. and Gavrila, D. M. (2009). Monocular pedestrian detection: Survey and experiments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(12):2179–2195.

Faouzi, N.-E. E., Leung, H., and Kurian, A. (2011). Data fusion in intelligent transportation systems: Progress and challenges - a survey. *Information Fusion*, 12(1):4–10.

Felzenszwalb, P. F., Girshick, R. B., and McAllester, D. (Accessed on July, 2012). Discriminatively trained deformable part models, release 4. http://people.cs.uchicago.edu/ pff/latent-release4/.

Felzenszwalb, P. F., Girshick, R. B., 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. *Intelligent Transportation Systems, IEEE Transactions on*, 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 *Information Fusion (FUSION), 2011 Proceedings of the 14th International Conference on*, pages 1 –8.

Geronimo, D., Lopez, A. M., Sappa, A. D., 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, H. P., Cosatto, E., Bottou, L., Durdanovic, I., and Vapnik, V. (2005). Parallel support vector machines: The cascade svm. In *In Advances in Neural Information Processing Systems, NIPS*, pages 521–528.

Heikkila, J. and Silven, O. (1997). A four-step camera calibration procedure with implicit image correction. In *Computer Vision and Pattern Recognition, CVPR. IEEE Conference on*, pages 1106–1112.

Kang, P. and Cho, S. (2006). Eus svms: Ensemble of under-sampled svms for data imbalance problems. In *Neural Information Processing*, volume 4232 of *Lecture Notes in Computer Science*, pages 837–846. Springer Berlin / Heidelberg.

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 *Intelligent Transportation Systems, ITSC. IEEE International Conference on*, pages 1574–1579, USA.

Markoff, J. (Oct 9, 2010). Google cars drive themselves, in traffic. New York Times.

Navarro-Serment, L. E., 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.

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, F. P., 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. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*.

Viola, P. and Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. In *IEEE Computer Vision and Pattern Recognition (CVPR 01)*, volume 1, pages 511–518, Hawaii,USA.