LIDAR and Vision-Based Pedestrian Detection System

Cristiano Premebida, Oswaldo Ludwig, and Urbano Nunes
Institute for Systems and Robotics
Department of Electrical and Computer Engineering
University of Coimbra
Coimbra 3030-290, Portugal
e-mail: cpremebida@isr.uc.pt, oludwig@isr.uc.pt, urbano@isr.uc.pt

Received 11 September 2008; accepted 30 July 2009

A perception system for pedestrian detection in urban scenarios using information from a LIDAR and a single camera is presented. Two sensor fusion architectures are described, a centralized and a decentralized one. In the former, the fusion process occurs at the feature level, i.e., features from LIDAR and vision spaces are combined in a single vector for posterior classification using a single classifier. In the latter, two classifiers are employed, one per sensor-feature space, which were offline selected based on information theory and fused by a trainable fusion method applied over the likelihoods provided by the component classifiers. The proposed schemes for sensor combination, and more specifically the trainable fusion method, lead to enhanced detection performance and, in addition, maintenance of false-alarms under tolerable values in comparison with single-based classifiers. Experimental results highlight the performance and effectiveness of the proposed pedestrian detection system and the related sensor data combination strategies.

1. INTRODUCTION

Intelligent ground vehicles as well as mobile robots, navigating in environments with static and dynamic objects around, e.g., other vehicles, mobile robots, and vulnerable road users (particularly pedestrians), should be provided with perception systems whose primary function is to detect and classify surrounding objects, having in view to avoid collisions and to mitigate situations of risk during the navigation. A key element of a perception system can be a single and reasonable cost-affordable sensor or, on the other hand, a set of multiple sensors for providing data to higher decision levels in charge of performing classification and/or situation assessment. The complementary and redundant information that can be obtained using a multisensor architecture should be properly explored to maximize the inference and confidence levels in object detection, which constitute a prerequisite for a complete pedestrian protection system.

The integration of a LIDAR (LIght Detection And Ranging) sensor and a camera, to bring redundancy and complementary characteristics for improving the detection system's reliability and accuracy, had gained the attention of the intelligent vehicles (IV)

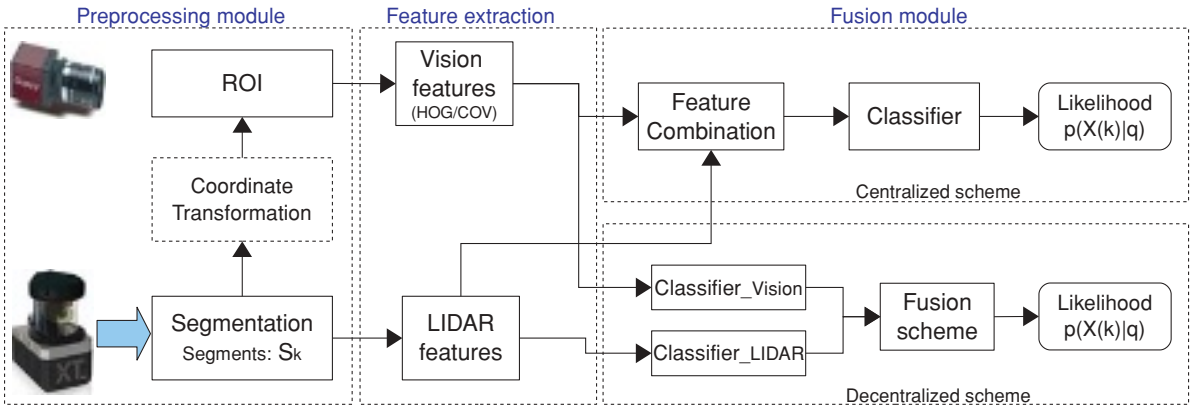


Figure 1. Sensor fusion architecture composed of three modules: preprocessing, feature extraction, and fusion.

and mobile robotics research communities in the past few years. Handling this multisensorial problem is not just a matter of determining regions of interest (ROI) in the image to perform vision-based classification; in fact, many important steps have to be properly addressed before a high-level combination can be achieved. This work attempts to contribute to the solution of the pedestrian detection problem using an ensemble of classifiers fusing LIDAR and vision data.

This paper presents research results on centralized and decentralized schemes proposed to combine range and visual information, gathered by a Ibeo Alasca-XT LIDAR and an Allied Guppy camera setup, mounted in an electrical vehicle (detailed in Section 2), with the main goal of performing pedestrian detection in outdoor urban-like scenarios. The proposed system is composed of three main modules: preprocessing, feature extraction, and fusion modules (see Figure 1).

The preprocessing module, presented in Section 2, is in charge of data acquisition, segmentation (in the laser space), and ROI determination in the image plane. Basically, this module generates the entities/objects of interest for further classification. The feature extraction module, detailed in Section 3, calculates two categories of features: a 15-dimensional laser-based feature vector and the image-based feature array composed of histogram of oriented gradients (HOG) and covariance (COV) features. The fusion module involves the classification methods, described in Section 4, and the fusion architectures: centralized and decentralized. In the former, the best classifier, applied over the whole feature space, is used to perform the final inference, whereas in the latter the likelihoods provided by two classifiers, one per feature space, are combined through a set of fusion methods; the pair of classifiers was selected based on a maximum relevance and minimal redundancy criterion (mRMR) (Peng, Long, & Ding, 2005). Finally, the fusion schemes, categorized as trainable fusion methods and nontrainable fusion rules, are outlined in Section 5.

The proposed fusion schemes for pedestrian detection were validated using our data set, which was previously separated in training and testing parts, each part corresponding to different conditions under which the data were collected. Experimental results are reported and analyzed in Section 6, with emphasis on the results obtained with the proposed trainable fusion methods. Finally, conclusions are drawn in Section 7.

Table I surveys some significant related works on pedestrian detection using LIDAR and monocular visible-spectrum cameras. Other relevant related works are Gandhi and Trivedi (2007) on pedestrian protection systems and Hall and Llinas (1997) on multisensor data fusion.

This paper makes some contributions within the pedestrian detection theme, which are mainly threefold:

1. LIDAR-based classifiers: A set of consistent methods is used to classify pedestrians using a feature vector with 15 components, some of them proposed in this work.

Table I. Survey of some related work on vision and LIDAR–based perception systems for pedestrian and on-road object detection/classification in outdoor scenarios.

Ref.	Vision system	LIDAR system	Comments
Douillard et al., 2007	Monocular color camera. Conditional random fields (CRFs) trained with virtual evidence boosting (VEB).	Single-layer LIDAR. Geometrical information is processed from the LIDAR data to estimate/classify the objects as vehicles or nonvehicles.	To deal with the problem of object scale variations in the images, the range information provided by the LIDAR is used during the CRF classification. The classification method was evaluated and compared in several features: geometrical (from laser data), visual (color and texture), and combination of both. The CRF and a LogitBoost classifier were also compared.
Spinello and Siegwart, 2008	HOG–SVM classifier based on monocular color images.	A multilayer LIDAR (Ibeo Alasca XT) is employed to detect on-road objects whose positions are projected into the image plane.	The object's position is detected by the LIDAR, and the vision-based system classifies the detected objects as pedestrian or nonpedestrian. A Bayesian decomposed expression is used as the reasoning fusion rule.
Pangop, Chapuis, Bonnet, Cornou, and Chausse, 2008	An Adaboost classifier, trained with Haar-like features, is used to classify pedestrians.	An Ibeo Alasca XT LIDAR is employed for object segmentation, tracking, and detection.	The speed, estimated during the tracking process, and the vision score–based likelihood are fused in a Bayesian framework using an autoregressive (AR) formalism to model the observations.
Hwang, Cho, Ryu, Park, and Kim, 2007	Monocular color camera. A multiple-SVM classifier is used to verify the hypothesis candidates inside the ROIs.	Single-layer LIDAR. The entities detected by the LIDAR generate hypothesis candidates, which are projected in the image plane (ROI) by means of perspective mapping.	The image plane is subdivided into five areas, where different trained SVMs are employed to classify the vehicles. A comparison study between single-SVM and five-SVM approaches is presented.
Mahlisch, Schweiger, Ritter, and Dietmayer, 2006	Monocular color camera. An Adaboost, using Haar-like features, processes the images delimited by the ROIs.	Multilayer LIDAR. The objects detected by the LIDAR define the ROI in the image plane.	The paper is focused on the newly proposed method designated "cross-calibration." The idea behind this method is to facilitate the correspondence between the LIDAR space and the image plane projection.
Szarvas, Sakai, and Ogata, 2006	Monocular gray-scale camera. A convolutional NN classifier is used.	Multilayer LIDAR. Objects detected in the LIDAR space are projected to the image plane (ROI) using perspective mapping (intrinsic–extrinsic parameters are obtained).	A relaxed flat world is used to model the road. Some comparisons are presented considering some variations of the system: vision, vision and LIDAR–based ROI, flat model, and nonrestricted road model.
Cheng, Zheng, Zhang, Qin, and van de Wetering, 2007	Two monocular color cameras: one camera for lane detection and the other for vehicle detection using Gabor features.	Single-layer LIDAR and radar. Using an extended Kalman filter (EKF), local tracking techniques are used in LIDAR and the radar reference system and fused to form a global tracking approach.	The fusion strategy using LIDAR and radar information, for on-road object detection, constitutes the focus of this paper, with emphasis on a local and global tracking approach. The vision-based obstacle detection system uses range information available from global tracks, in the form of ROI, as a decision-making system.

(a) Electric vehicle used in the experiments

(b) Sensors: camera and LIDAR setup

Figure 2. Electric vehicle and the sensors used in data set acquisition.

2. Mutual-information-based classifier selection method: A classifier selection approach based on maximum relevance and minimal redundancy is proposed here as an attempt to obtain an "optimal" classifier ensemble, avoiding brute-force selection methods.
3. Trainable fusion methods: A set of trainable fusion methods is used here to fuse the selected classifiers. The trainable fusion outperformed the nontrainable-based fusion rules.

We have made our data set available online[1] for further comparison studies and public usage. It is important to clarify that those contributions are still ongoing approaches that will be further explored and improved bearing in mind feasible and reliable pedestrian protection systems.

2. PREPROCESSING MODULE

The LIDAR used in our system is the Ibeo Alasca-XT, a four-layer laser scanner that was mounted in a "rigid" platform on the frontal part of the vehicle, working at 12.5 Hz (scans per second). The data stream is sent to the host laptop by means of an Arcnet-PCMCIA adapter, and the acquisition algorithm is based on the Ibeo Linux-API. The acquired scans consist of raw range data that are treated as clouds of points.

The second sensor in use is an Allied Guppy camera, with Bayer-type sensor and IEEE 1394 protocol. The images are acquired using openCV-based libraries in a sequential way, having the Ibeo API thread priority over the process. The images were transformed to red–green–blue (RGB) standard for offline processing purpose, i.e., for feature extraction and pedestrian detection.

The data set has been recorded in the Institute for Systems and Robotics–University of Coimbra (ISR-UC) Campus[2] open surrounding areas, with many static and moving pedestrians and cars around, using the vehicle, driven manually, and the sensor apparatus shown in Figure 2.

For each scan delivered by the LIDAR, some preprocessing tasks have to be processed in advance before the calculation of the feature vectors and the subsequent object classification. The tasks performed in the LIDAR preprocessing module are prefiltering, coordinate transformation, and segmentation.

Prefiltering is applied to filter the incoming raw data in order to detect isolated/spurious range points, discarding measurements that occur out of a predefined field of interest, and to perform pertinent data processing that leads to decreased complexity and processing time of subsequent stages. Coordinate transformation, in our case, is a conversion from polar to Cartesian coordinates. The segmentation stage constitutes a critical part in such

[1]http://www.isr.uc.pt/~cpremebida/dataset.

[2]http://www.isr.uc.pt/~cpremebida/PoloII-Google-map.pdf.

perception systems and can be performed by means of specific methods as presented in Premebida and Nunes (2005), Spinello and Siegwart (2008), and Streller and Dietmayer (2004).

For allowing a better generalization of the methods, all the range data are considered as two-dimensional (2D) measurements; for multilayer laser scanners (e.g., Ibeo Alasca), the scanned points are projected to a single reference plane.

Expressing a 2D full scan as a sequence of N_S measurement points in the form $\text{Scan} = \{(r_l, \alpha_l) | l = 1, \ldots, N_S\}$, where (r_l, α_l) denotes the polar coordinates of the lth *scan point*, a group of scan points that constitute a segment S_k can be expressed as

$$S_k = \{(r_n, \alpha_n)\}, \quad n \in [l_i, l_f], \quad n = 1, \ldots, \text{np}, \qquad (1)$$

where np is the number of points in the current segment and l_i and l_f are the initial and the final scan points that define the segment. A segment can also be defined in Cartesian coordinates $\mathbf{x} = (x_k, y_k)$, where $(x_k = r_n \cos \alpha_n, \quad y_k = r_n \sin \alpha_n)$. Henceforth, a *segment* is explicitly related to a group of range points related to one, unambiguously, object of interest and expressed by S_k.

It is important to mention that to use a multilayer laser conveniently, each layer has to be processed separately, especially to avoid false alarms due to pitch oscillations or road inclinations, and when the vehicle is driving on irregular roads. Furthermore, as we have used the raw data (i.e., without the Ibeo processing unit), we faced another problem: the acquired data came in a nonordered sequence, forcing the usage of some additional processing steps to separate the vertical layers properly and to order the data.

As this paper is mainly focused on the fusion and combination of LIDAR and vision data for pedestrian detection, all the extracted segments S_k that constitute our data set were validated under user supervision to guarantee that each laser segment represents unambiguously a single object (positive or negative). It is relevant to note that in realistic situations some problems invariably will arise, such as data association errors, oversegmentation, missing measurements, and tracking inconsistencies.

On the other hand, the images extracted from the ROIs in the image plane were not postprocessed; this means that all the cropped images in the data set were extracted automatically from ROIs obtained using LIDAR segments projected in the image plane and, as a consequence, are prone to error due to calibration imprecision, road irregularities, vehicle vibrations, and so on. Nevertheless, we decided to allow those cropped images with no user intervention or any correction, resulting in a closer realistic image-based data set.

The calibration procedure is necessary to obtain a mapping to transform points in the laser reference system $\{L\}$ to the camera reference system $\{C\}$ and then to the image plane. In the calibration process it was considered that both sensors were stable and that the mechanical vibrations and oscillations were negligible. Using a flat target ("checkerboard"), positioned at different distances from the laser–camera setup, the transformation between $\{L\}$ and $\{C\}$ was obtained under a quadratic error minimization criterion using the method proposed by Zhang and Pless (2004). A set of images and laser measurements taken at different positions of the target were used to estimate the coordinate transformation and also the camera's intrinsic and extrinsic parameters.

With the LIDAR data it is possible to obtain only the horizontal limits of the object position in 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 the maximum height for a pedestrian.

The following estimated matrix, necessary to make a rigid correspondence between the laser scanner and the camera reference system, was obtained:

$$T_C^L = \begin{bmatrix} 0.99986 & -0.014149 & -0.0093947 & 11.917 \\ 0.014395 & 0.99954 & 0.026672 & -161.26 \\ 0.009013 & -0.026804 & 0.9996 & 0.77955 \end{bmatrix}, \qquad (2)$$

where the translational vector components are in millimeters. The extrinsic and intrinsic pinhole camera model, as well as the scripts with all the pertinent variables necessary to accomplish Eq. (2), are available online.

3. FEATURE EXTRACTION

A 15-dimensional LIDAR-based feature vector and the well-known HOG and COV image descriptors are addressed in the next subsections.

Table II. LIDAR features for pedestrian classification.

f_i	Formula	Comments
$f1$	$\text{np} \cdot r_{\min}$	The product of the number of range points (np) with the minimum range distance ($r_{\min}$).
$f2$	np	Number of points. This "simple" feature is here considered just for comparison purposes.
$f3$	$\sqrt{\Delta X^2 + \Delta Y^2}$	Normalized Cartesian dimension: this feature corresponds to the root mean square of the segment width (ΔX) and length (ΔY) dimensions.
$f4$	$\sqrt{\frac{1}{\text{np}} \sum_{n=1}^{\text{np}} \|\mathbf{x}_n - \bar{\mathbf{x}}\|}$	Internal standard deviation: denotes the standard deviation of the range points ($\mathbf{x}_n$) with respect to the segment centroid $\bar{\mathbf{x}}$.
$f5$	Radius $\leftarrow$ fitted circle	Radius: denotes the radius of a circle extracted from the segment points. Guivant's method (Guivant, Masson, & Nebot, 2002) was used in fitting the circle and to extract the corresponding radius.
$f6$	$\frac{1}{\text{np}} \sum_{n=1}^{\text{np}} \|\mathbf{x}_n - \tilde{\mathbf{x}}\|$	Mean average deviation from the median $\tilde{\mathbf{x}}$.
$f7$	IAV	The inscribed angle variance (IAV), proposed by Xavier, Pacheco, Castro, Ruano, and Nunes (2005), corresponds to the mean of the internal angles along the extreme points and the in-between points that constitute the segment.
$f8$	std($f7$)	Standard deviation of the inscribed angles calculated previously.
$f9$	$\frac{1}{\text{np}} \sum_{n=1}^{\text{np}} (\mathbf{x}_n - \hat{\mathbf{x}}_{l,n})^2$	Linearity: this feature measures the straightness of the segment and corresponds to the residual sum of squares to a line $\hat{\mathbf{x}}_{l,n}$ fitted into the segment in the least-squares sense.
$f10$	$\frac{1}{\text{np}} \sum_{n=1}^{\text{np}} (\mathbf{x}_n - \hat{\mathbf{x}}_{c,n})^2$	Circularity: this feature measures the circularity of a segment. Like for the $f9$ feature, we sum up the squared residuals to a fitted circle $\hat{\mathbf{x}}_{c,n}$.
$f11$	$\sum_{n=1}^{\text{np}} \frac{(\mathbf{x}_n - \mu_{\mathbf{x}})^{ko}}{\text{np}}$	Second central moment: it is the second moment taken about the mean $\mu_{\mathbf{x}}$, where ko is order of the moment, i.e., $ko = 2$.
$f12$	—	Third central moment, $f11$ with $ko = 3$.
$f13$	—	Fourth central moment, $f11$ with $ko = 4$.
$f14$	$\sum_{n=1}^{\text{np}} \|\mathbf{x}_n - \mathbf{x}_{n-1}\|$	Segment length: this feature is defined as the summation over the norm of the Euclidean distance between adjacent points.
$f15$	std($f14$)	Standard deviation of the segment length.

3.1. LIDAR-Based Features

Feature extraction from LIDAR data and its utilization for pedestrian detection in urban environments is a subject that has not been investigated significantly, although some works are worthy of mention: Douillard, Fox, and Ramos (2007), Premebida and Nunes (2006), and Streller and Dietmayer (2004). Nevertheless, in the mobile robotics field, the work by Arras, Mozos, and Burgard (2007) is a reference on using purely LIDAR features[3] for human detection in indoor environments. The components of the laser-based feature vector used in the present work, many of them based on Arras's work, are detailed in Table II.

[3]The object speed could be considered an exception.

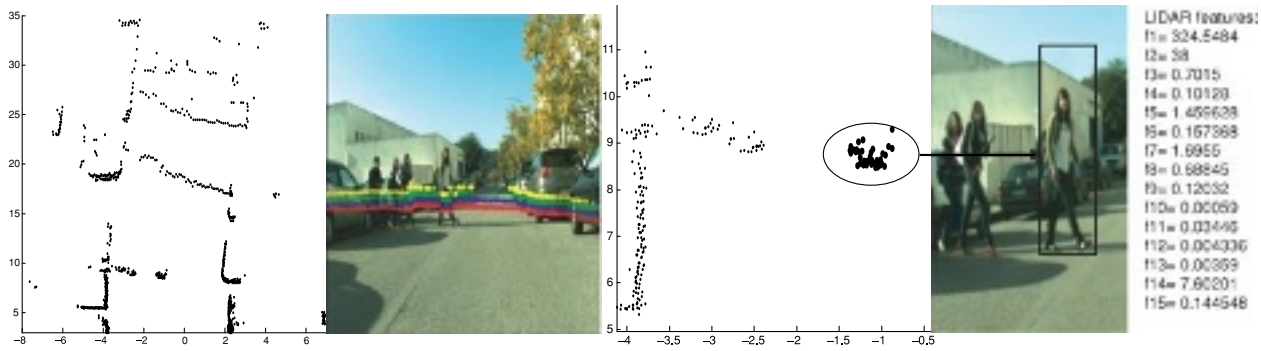


Figure 3. An example that illustrates a pedestrian perceived by the laser, as a segment of range points, with its ROI in the image plane and the corresponding laser features. The images are depicted to facilitate understanding of the scene.

The feature vector extracted from a segment S_k is calculated using only 2D information in polar and/or Cartesian space; hence as said previously for the case of a multilayer LIDAR, the "vertical" information has to be projected on a common 2D plane, which means that all these features can be used in the case of single-layer lasers. Figure 3 illustrates range readings from a scene where a pedestrian, its corresponding segment, and the image ROI are highlighted as well as the related laser features.

3.2. HOG Features

HOG descriptors (Dalal & Triggs, 2005) are reminiscent of edge-oriented histograms, scale-invariant feature transform (SIFT) descriptors (Lowe, 2004), and shape contexts. To compose HOG, the cell histograms of each pixel within the cell cast a weighted vote, according to the gradient $L2$-norm, for an orientation-based histogram channel. In this work the histogram channels are calculated over rectangular cells (i.e., R-HOG) by the computation of unsigned gradient. The cells overlap half of their area, meaning that each cell contributes more than once to the final feature vector. To account for changes in illumination and contrast, the gradient strengths were locally normalized, i.e., normalized over each cell. The HOG parameters were adopted after a set of experiments performed over the training data set using a neural network (NN) as classifier. The highest area under the receiver operating characteristic (ROC) curve (AUC), computed over the validation data set, was achieved by means of nine rectangular cells and nine bin histograms per cell. The nine histograms with nine bins were then concatenated to make a 81-dimensional feature vector.

3.3. COV Features

The utilization of covariance matrix descriptors in classification problems was followed by Tuzel, Porikli, and Meer (2006, 2007). Let I be the input image matrix and z_p the corresponding d-dimensional feature vector calculated for each pixel p:

$$z_p = \left[x, y, |I_x|, |I_y|, \sqrt{I_x^2 + I_y^2}, |I_{xx}|, |I_{yy}|, \arctan\frac{|I_y|}{|I_x|}\right], \tag{3}$$

where x and y are the pixel p coordinates; I_x and I_y are the first-order intensity derivatives regarding x and y, respectively; I_{xx} and I_{yy} are the second-order derivatives; and the last term is the edge orientation.

In this work, four subregions are computed within a region R, which represents the area of a cropped image. Each subregion overlaps half of its area, meaning that each subregion contributes more than once to the final feature vector. For the ith rectangular subregion R_i, the covariance matrix C_{R_i} is expressed by

$$C_{R_i} = \frac{1}{N_i - 1}\sum_{p=1}^{N_i}(z_p - \mu)(z_p - \mu)^T, \tag{4}$$

where μ is the statistical mean of z_i and N_i is the number of pixels of the subregion R_i (in this case $i = 1 \ldots 4$). Notice that, due to the symmetry of C_{R_i}, only the upper triangle part needs to be stored, and hence the covariance descriptor of a subregion is an 8×8 matrix. The features of the whole region R are also calculated; therefore a feature vector with

180 features is generated, i.e., 4 subregions R_i, totaling 144 features, plus 36 features of the whole region R.

4. CLASSIFIERS

Five classifiers are discussed in this section: naive Bayes, GMMC, MCI–NN, FLDA, and RBF–SVM. These classifiers are used in two situations: as single classifiers and as the basis of trainable fusion methods.

4.1. Naive Bayes

Based on the assumption that each feature is statistically independent, the probability density function (pdf) that characterizes the object class m is modeled as the product of each feature-based pdf. A one-dimensional Gaussian $\theta_k(\mu_k, \sigma_k)$ was considered in modeling each pdf:

$$p(x_k|q_m, \theta_k) = \frac{1}{\sigma_k\sqrt{(2\pi)}} \exp\left[\frac{-(x_k - \mu_k)^2}{2\sigma_k^2}\right], \quad (5)$$

where μ_k is the mean and σ_k is the statistical variance for the kth feature x_k and q_m corresponds to the "object" class of interest, i.e., pedestrians q_1 and non-pedestrians q_2.

For the case q_1, the likelihood $\mathcal{L}$ is obtained by the normalization

$$\mathcal{L}(x_k|q_1) = \frac{p(x_k|q_1, \theta_k)}{p(x_k|q_1, \theta_k) + p(x_k|q_2, \theta_k)}, \quad (6)$$

and therefore, the combined likelihood is expressed by

$$\mathcal{L}(x|q_1) = \prod_{k=1}^{d} \mathcal{L}(x_k|q_1), \quad (7)$$

where x corresponds to a d-dimensional feature vector.

4.2. GMMC

For the GMMC classifier, the likelihood is calculated considering a mixture of M Gaussian pdf, defined by $\Theta_i(\rho_i, \mu_i, \Sigma_i)$, where ρ_i is a weight vector, such that $\sum_{i=1}^{M} \rho_i = 1$, μ_i is the d-dimensional mean vector, and Σ_i is the covariance matrix. The pdf for a single component i is modeled as

$$p(x|q_m, \Theta_i) = \frac{\exp\left[-\frac{1}{2}(x - \mu_i)^T(\Sigma_i)^{-1}(x - \mu_i)\right]}{\sqrt{(2\pi)^d}}. \quad (8)$$

Finally, the likelihood is the linear composition

$$\mathcal{L}(x|q_1) = \sum_{i=1}^{M} \rho_i \cdot p(x|q_1, \Theta_i). \quad (9)$$

4.3. MCI–NN

Minimization of interclass interference (MCI) (Ludwig and Nunes, 2008) is a maximum-margin–based training algorithm for NN. MCI aims to create a NN hidden layer output (i.e., feature space) in which the patterns have a desirable statistical distribution. Regarding the neural architecture, the linear output layer is replaced by the Mahalanobis kernel in order to improve generalization. MCI is applicable on a neural network model with two sigmoidal hidden layers and one output nonlinear layer:

$$yhf = \varphi(W_1 \cdot x + b_1), \quad (10)$$

$$yhs = \varphi(W_2 \cdot yhf + b_2), \quad (11)$$

$$\hat{y} = \frac{d_2 - d_1}{d_2 + d_1}, \quad (12)$$

where yhf is the output vector of the first hidden layer; yhs is the output vector of the second hidden layer; W_k $(k = 1, 2)$ is the synaptic weights matrix of the layer k; b_k is the bias vector of layer k; x is the input vector; $\varphi(.)$ is the sigmoid function; $d_m = (yhs - \mu_m)^T \Sigma^{-1}(yhs - \mu_m)$ is the Mahalanobis distance between yhs and μ_m; Σ is the covariance matrix over all the output vectors yhs, presented by the second hidden layer in response to the training data set; $\mu_m = \frac{1}{N_m}\sum_{i=1}^{N_m} yhs_m(i)$ is the prototype of class m; N_m is the number of training patterns that belong to class m; and $yhs_m(i)$ is the second hidden layer output for an input that belongs to class m. Analyzing Eq. (12), we can observe that $\hat{y}$ varies continuously from -1, for $yhs = \mu_2$, to 1, for $yhs = \mu_1$. This continuous approach enables ROC curve calculation.

The MCI creates a hidden space where the Euclidean distance between the prototypes of each class is increased and the pattern dispersion of each class is decreased. The goal is to maximize the objective

function

$$J = (\mu_1 - \mu_2)^T(\mu_1 - \mu_2) - \delta_1^2 - \delta_2^2, \tag{13}$$

where $\delta_m^2 = \sum_{i=1}^{N_m}[yh_m(i) - \mu_m]^T[yh_m(i) - \mu_m]$ is the deviation of class m patterns in the hidden space. The weights and biases are updated based on the gradient ascendant algorithm.

4.4. FLDA

Let us consider w a vector of adjustable gains and $\{x_c\}$ the set of feature vectors that belong to class c, $(c = 1, 2)$ with mean μ_c, and covariance $\sum_c$. The linear combination $w \cdot x_c$ has mean $w \cdot \mu_c$ and covariance $w^T \Sigma_c w$. The ratio, $J(w)$, of the variance *between* the classes, $\sigma_{\mathbf{b}}^2$, by the variance *within* the classes, $\sigma_{\mathbf{w}}^2$, is a suitable measure of separation between these two classes:

$$J(w) = \frac{\sigma_{\mathbf{b}}^2}{\sigma_{\mathbf{w}}^2} = \frac{[w \cdot (\mu_2 - \mu_1)]^2}{w^T(\Sigma_1 + \Sigma_2)w}. \tag{14}$$

To obtain the maximum separation between classes, one has to find the vector w that solves the optimization problem:

$$\max_w J(w), \tag{15}$$

whose solution is

$$w = (\Sigma_1 + \Sigma_2)^{-1}(\mu_2 - \mu_1). \tag{16}$$

To find the plane that best separates the data, $w^T\mu_1 + b = -(w^T\mu_2 + b)$ has to be solved for the bias b.

4.5. SVM

Support vector machines (SVM) are based on the statistical theory of learning, developed by Vapnik (1998). This theory provides a set of principles to be followed in order to obtain classifiers with good generalization, defined as its ability to predict correctly the class of new data in the same area where the learning occurred. Table III presents three usual SVM kernels, where nd is a natural number denoting the polinomial degree.

Table III. Usual SVM kernels.

Kernel name	Kernel function
Linear	$H(x, x') = x^T x'$
Polynomial	$H(x, x') = (x^T x' + 1)^{\text{nd}}$
RBF	$H(x, x') = \exp(-\gamma \|x - x'\|^2)$

SVM is very sensitive to the margin parameter C,[4] and therefore it is not appropriate to adjust this parameter based on the SVM performance on the test data set; otherwise we will bring information from the test data set to the SVM. The usual approach is to apply K-fold cross validation over the training data set.

5. FUSION MODULE

In this section we will cover the following subjects: nontrainable rules and trainable fusion techniques, centralized and decentralized schemes, and, finally, a classifier selection approach.

5.1. Nontrainable Fusion Rules

A fusion strategy is necessary to combine information from each classifier in order to provide a final classification reasoning. The likelihoods $\mathcal{L}_i$ yielded by each classifier Θ_i, $(i = 1, \text{nc})$, where nc is the number of classifiers, are fused by three nontrainable fusion rules: average $\mathcal{F}_{\text{Avage}}(\Theta_i)$, maximum $\mathcal{F}_{\text{Max}}(\Theta_i)$, and naive-product $\mathcal{F}_{\text{Nprod}}(\Theta_i)$ (naive Bayes inspired). The former two are simple and intuitive rules, however important for comparative purposes, and the latter deserves some explanation.

Considering the recursive Bayesian updating approach, the joint probability of the class being a pedestrian (q_1) is computed as

$$P(q_1|\Theta_1, \ldots, \Theta_{\text{nc}}) = \frac{P(q_1)P(\Theta_1|q_1)P(\Theta_2|\Theta_1 \ldots \Theta_{\text{nc}})}{P(\Theta_1, \ldots, \Theta_{\text{nc}})}. \tag{17}$$

[4]Margin parameter that determines the trade-off between maximization of the margin and minimization of the classification error (Abe, 2005).

Assuming classifiers' independence, Eq. (17) becomes

$$P(q_1|\Theta_1,\ldots,\Theta_{\text{nc}}) = \frac{P(q_1)\prod_{i=1}^{\text{nc}} P(\Theta_i|q_1)}{P(\Theta_1,\ldots,\Theta_{\text{nc}})}. \quad (18)$$

Equations (18) depend on the prior information $P(q_1)$ about the actual detected object being a pedestrian, which can be based on a predefined model or estimated during a tracking process. Under the strong assumption of $P(q_1) = P(q_2)$, the naive-product rule becomes

$$\mathcal{F}_{\text{Nprod}} = \prod_{i=1}^{\text{nc}} \mathcal{L}_i. \quad (19)$$

5.2. Trainable Fusion Methods

The trainable fusion algorithm (Ludwig, Delgado, Gonçalves, & Nunes, 2009) (represented by the fusion scheme block in Figure 1) is also a classifier that receives the likelihoods from the single classifiers (Classifier_Vision and Classifier_LIDAR in Figure 1) and outputs the likelihood of the classifier ensemble (decentralized-fusion scheme case). In this work, five trainable fusion algorithms were tested, each one corresponding to one of the five classifiers described in Section 4. All the single classifiers and the trainable fusion algorithms are trained with the same training data set. However, the single classifiers are trained before, in order to create a *likelihood training data set* $\{\mathcal{U}_{\text{train}}\}$, which is used together with the training labels $\{y_{\text{train}}\}$ in the fusion algorithm training process. Algorithm 1 details the training process of the fusion classifier.

5.3. Centralized Fusion Scheme

In this type of fusion scheme the fusion occurs at the feature level, i.e., the LIDAR and the vision-based features are combined in a single vector. The classifiers are trained with all the available features, and the best classifier is chosen to assess the final object classification.

In our case, three single classifiers, FLDA, RBF–SVM, and MCI–NN, have been trained with the complete feature set (*all-features* vector with 276 components) and the classifier selection has been done considering the accuracy (Acc) and the balanced error rate (BER), both of them calculated over the training data set.

Algorithm 1 Training process of the fusion methods

Input: $\{x_{\text{train}}\}$, $\{y_{\text{train}}\}$: training data set and the ground-truth labels
nc: number of single classifiers
NF: number of fusion methods (NF = 5)
ns: number of training examples
Θ_k: set of single classifiers; $(k = 1\cdots\text{nc})$
Output: set of trained model $\{\mathcal{F}_k\}$; $(k = 1\cdots NF)$
1: $\mathcal{U}_{\text{train}} \leftarrow$ empty matrix;
2: **for** k = 1:nc **do**
3: process $\{x_{\text{train}}\}$ through the previous trained classifier Θ_k, to obtain the ns-dimensional likelihood vector $\mathcal{L}^k_{\text{train}}$;
4: $\mathcal{U}_{\text{train}} \leftarrow [\mathcal{U}_{\text{train}}|\mathcal{L}^k_{\text{train}}]$: concatenate the likelihood vector to create the likelihood training matrix;
5: **end for**
6: **for** k=1:NF **do**
7: apply $\mathcal{U}_{\text{train}}$ and $\{y_{\text{train}}\}$ to train the kth fusion method $\mathcal{F}_k$.
8: **end for**

5.4. Decentralized Fusion Scheme

In this fusion scheme, each classifier is supposed to be specialized in a part of the feature set. In the present case, it is a straightforward decision to separate the feature space in LIDAR and vision-based subspaces (see Figure 1). The five single classifiers, described in Section 4, were used in the LIDAR space. As concerns vision space, FLDA, RBF–SVM, and MCI–NN classifiers were employed. To avoid singularities on the covariance matrices and/or likelihoods tending to zero, GMM and naive-Bayes classifiers were discarded.

A key issue is to select the pair of single classifiers, aiming to have one *expert* in LIDAR space and the other *expert* in vision space. Instead of using heuristics or empirical approaches for the classifier selection process, we propose to use a method based on information theory, explained in the next section.

The classifier selection criterion is based on the principle of minimal-redundancy-maximal-relevance (mRMR) (Peng et al., 2005). Therefore, considering the random variable $\mathcal{L}_i$ as the likelihood of classifier i and Y as the respective target output (label), the relevance V of a set of nc classifiers is the mean value of the mutual information $I(\mathcal{L}_i; Y)$ between the classifier likelihood and the labels:

$$V = \frac{1}{\text{nc}} \sum_{i=1}^{\text{nc}} I(\mathcal{L}_i; Y), \quad (20)$$

Table IV. Data sets used to train and to evaluate the classifiers.

Designation	N frames	N pos.	N neg.	Description
		Training data set		
ISR − UC_train	1,100	550	550	Sunny day, winter, collected between 15:30 and 16:30
		Testing data set		
ISR − UC_test	1,400	400	1,000	Sunny day, winter, collected between 12:00 and 17:30

and the redundancy P is the mean value of the mutual information $I(\mathcal{L}_i;\mathcal{L}_j)$ among classifier outputs:

$$P = \frac{1}{\mathrm{nc}^2}\sum_{i=1}^{\mathrm{nc}}\sum_{j=1}^{\mathrm{nc}} I(\mathcal{L}_i;\mathcal{L}_j). \tag{21}$$

The application of the mRMR principle corresponds to searching a set of classifiers to satisfy the maximization problem

$$\max_{\mathcal{L}_1\ldots\mathcal{L}_{\mathrm{nc}}} \Phi, \tag{22}$$

where $\Phi = V - P$. The solution of expression (22) is attained when the classifier likelihoods $\mathcal{L}_i$ are mutually exclusive and totally correlated to the target output Y. In other words, the idea is to take advantage of the classifier diversity.

6. EXPERIMENTAL RESULTS

The proposed detection system was evaluated in terms of Acc, AUC, BER, and ROC curves. Data sets, employed in the training and evaluation of the classifiers, are summarized in Table IV. The *ISR − UC_train* and *ISR − UC_test* data sets were acquired in the ISR–UC Campus, under the following configuration: the LIDAR field of view (FOV) was restricted to 180 deg, with a horizontal angular resolution of 0.5 deg, vertical resolution of [−1.2 deg, −0.4 deg, 0.4 deg, 1.2 deg], and measurement range up to 30 m; the camera FOV was 67 deg approximately. The data sets and the corresponding ground truth, generated under user supervision, are available online (http://www.isr.uc.pt/~cpremebida/dataset).

The training and the testing data sets were collected on different dates. Although both data sets were acquired around the same area, the positives (pedestrians) and the negatives are clearly different; another relevant aspect is that on the testing part, some samples were acquired at dusk, when the illumination condition changed drastically. Some images of the data sets are shown in Figure 4.

6.1. Centralized Fusion Scheme: Feature-Level Fusion

The centralized fusion scheme, in which all the features are concatenated in a single vector, was tested with the FLDA, RBF–SVM, and MCI–NN classifiers. Table V summarizes the results obtained by the single classifiers and the classifier ensemble with the centralized fusion structure. Regarding single classifiers, the naive-Bayes classifier had the best performance in the LIDAR space and the FLDA achieved the best scores in the vision feature space. Regarding the centralized scheme, the FLDA obtained the best results. ROC curves of the best single classifier and of the FLDA-based centralized fusion classifier are depicted in Figure 5. Additionally, Figure 6 illustrates some missing detections using single classifiers in the LIDAR and vision feature spaces, giving some insight about the missing occurrences.

Accuracy, BER, and AUC performance metrics are used throughout the paper. It is important to keep in mind that such metrics are calculated over all samples presented on the testing data set; hence those scores serve as a global indicator of the classifier performance. As a suggestion, we have selected a metric based on a useful percentile of the AUC, up to 10% of false positives, named $\mathrm{AUC}_{10\%}$, because the *optimal* operator point for ROC curves often does not occur out of that interval.

6.2. Classifier Selection for the Decentralized Fusion Scheme

Intuitively, the combination of the best LIDAR classifier with the best vision classifier would result in

Figure 4. Some samples to illustrate the different conditions and situations in which the data set has been acquired.

Table V. Performance metrics: testing data set.

	FLDA	Naive–Bayes	GMMC	RBF–SVM	MCI–NN
			LIDAR-based features		
Acc	0.786	0.883	0.875	0.840	0.861
BER	0.154	0.108	0.130	0.204	0.124
$AUC_{10\%}$	0.128	0.476	0.464	0.370	0.336
$TPr_{10\%}$	0.304	0.835	0.842	0.688	0.765
			Vision-based features		
Acc	0.846	—	—	0.841	0.813
BER	0.172	—	—	0.236	0.157
$AUC_{10\%}$	0.178	—	—	0.314	0.475
$TPr_{10\%}$	0.63	—	—	0.60	0.65
			Fused features (all features): centralized scheme		
Acc	0.880	—	—	0.846	0.810
BER	0.115	—	—	0.231	0.226
$AUC_{10\%}$	0.237	—	—	0.324	0.198
$TPr_{10\%}$	0.80	—	—	0.60	0.60

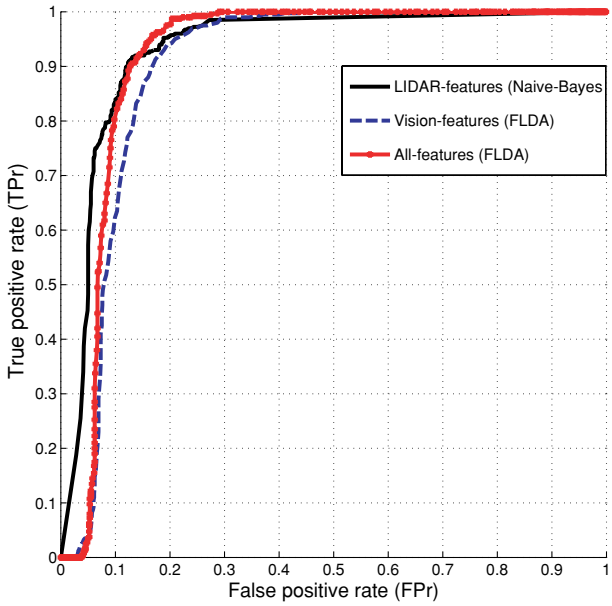


Figure 5. ROC for the best single classifiers and for FLDA-based centralized fusion scheme.

the optimal approach. However, the diversity among classifiers must be taken into account, i.e., redundancy has to be avoided in order to achieve better accuracy during the fusion process. According to the mRMR criterion, the fusion of the classifier GMMC, in the LIDAR space (LIDAR–GMM) with the FLDA in the vision space (vision–FLDA), is the "optimal" option, with $\Phi = -0.1665$. The value of Φ can be calculated by expressions (20)–(22) using $nc = 2$ and the values of mutual information, highlighted in bold, in Tables VI and VII. On the other hand, the fusion of the best LIDAR classifier (naive-Bayes) with the best vision classifier (FLDA) had a value of Φ ($\Phi = -0.1885$) lower than the previous pair combination. Experimental results reinforce the mRMR criterion, because the best accuracy of 89.92% was achieved by the trainable fusion scheme LIDAR–GMMC/vision–FLDA using the $\mathcal{F}_{\text{GMM}}$ fusion method (see Table VIII), whereas the classifier ensemble using the "intuitive best" set of single classifiers (LIDAR–naive-Bayes and vision–FLDA) achieved a maximum accuracy of 89.78%, below that of the former combination.

Figure 6. Some examples of miss detections using single classifiers: the LIDAR–naive and vision–LDA.

Table VI. Mutual information among classifiers (redundancy).

	LIDAR feature space					Vision feature space		
	LDA	Naive	GMM	SVM	NN	LDA	SVM	NN
LIDAR								
LDA	1.000	0.602	0.621	0.585	0.615	0.574	0.590	0.570
Naive	0.602	1.000	0.516	0.593	0.528	0.529	0.529	0.477
GMM	0.621	0.516	**1.000**	0.625	0.609	**0.661**	0.661	0.608
SVM	0.585	0.593	0.625	1.000	0.642	0.598	0.668	0.697
NN	0.615	0.528	0.609	0.642	1.000	0.652	0.653	0.570
Vision								
LDA	0.574	0.529	**0.661**	0.598	0.652	**1.000**	0.710	0.622
SVM	0.590	0.529	0.661	0.668	0.653	0.710	1.000	0.715
NN	0.570	0.477	0.608	0.697	0.570	0.622	0.715	1.000

Table VII. Mutual information between classifiers and the target output (relevance).

Feature space	Target output
LIDAR	
LDA	0.498
Naive	0.466
GMM	**0.642**
SVM	0.502
NN	0.580
Vision	
LDA	**0.686**
SVM	0.615
NN	0.521

6.3. Decentralized Fusion Scheme: Classifier Fusion

The proposed decentralized fusion architecture was tested with different fusion methods, trainable and nontrainable ones, using the classifiers with maximum relevance and minimal redundancy estimated over the training data set (LIDAR–GMMC and vision–FLDA).

Here, the most demanding task is the selection of the 15 possible combinations, five classifiers in the LIDAR space and three in the vision space. This aspect reinforces the need to avoid heuristic, or force-brute, methods and to consider a mutual information-based approach to aid in such labor, as described in the preceding section.

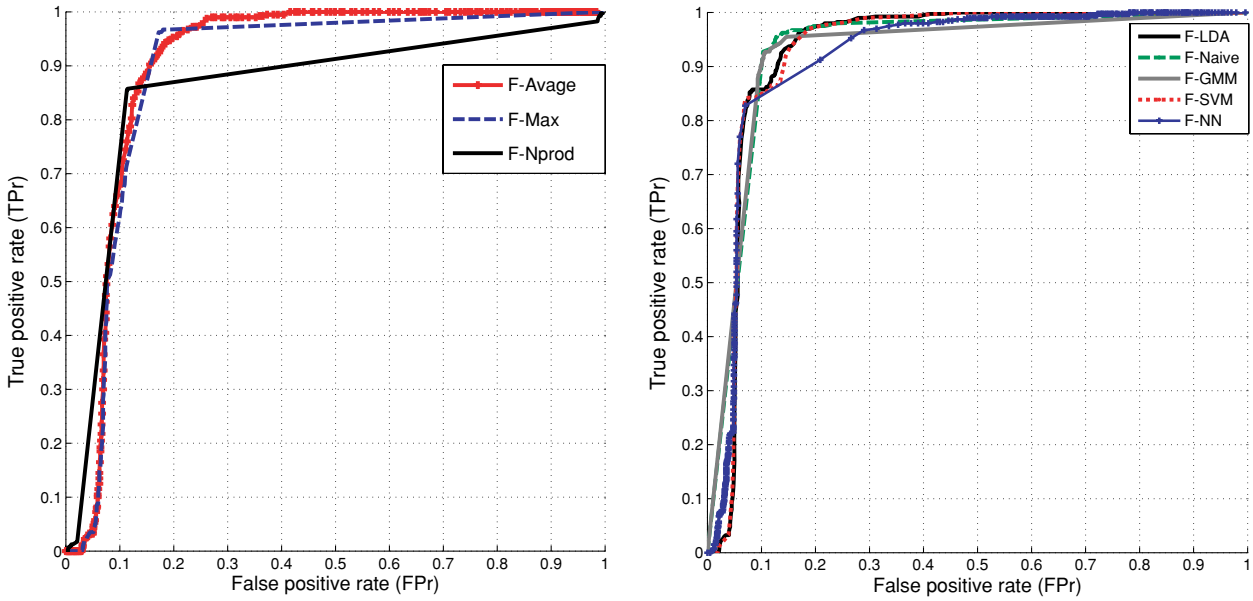
Once the most relevant and less redundant pair of classifiers has been selected, the set of proposed trainable fusion techniques, denoted $\mathcal{F}_{\text{LDA}}$, $\mathcal{F}_{\text{Naive}}$, $\mathcal{F}_{\text{GMM}}$, $\mathcal{F}_{\text{SVM}}$, and $\mathcal{F}_{\text{NN}}$, and the nontrainable rules, $\mathcal{F}_{\text{Avage}}$, $\mathcal{F}_{\text{Max}}$, and $\mathcal{F}_{\text{Nprod}}$, were applied over the testing data set. The results concerning each fusion technique are shown in Table VIII, and corresponding ROC curves are depicted in Figures 7(a) and 7(b).

7. CONCLUSION

Most of the works on pedestrian and on-road object detection using vision and LIDAR (see Table I for some relevant cases) employ a LIDAR to detect

Table VIII. Performance metrics for the fusion strategies: testing data set.

	Trainable methods					Nontrainable rules		
	$\mathcal{F}_{\text{LDA}}$	$\mathcal{F}_{\text{Naive}}$	$\mathcal{F}_{\text{GMM}}$	$\mathcal{F}_{\text{SVM}}$	$\mathcal{F}_{\text{NN}}$	$\mathcal{F}_{\text{Avage}}$	$\mathcal{F}_{\text{Max}}$	$\mathcal{F}_{\text{Nprod}}$
Acc	0.894	0.897	0.899	0.877	0.890	0.878	0.859	0.878
BER	0.109	0.108	0.096	0.124	0.102	0.122	0.109	0.122
$\text{AUC}_{10\%}$	0.391	0.460	0.465	0.412	0.421	0.282	0.178	0.197
$\text{TPr}_{10\%}$	0.861	0.892	0.912	0.855	0.850	0.684	0.633	0.710



(a) ROC: fusion with nontrainable rules

(b) ROC: fusion with trainable methods

Figure 7. The ROC using the decentralized fusion scheme.

objects and to generate target hypotheses inside the camera FOV, whereas a vision-based classifier is in charge of object validation or final classification. The present work contributes a sensor fusion strategy composed of centralized and decentralized fusion schemes. In the former the fusion process occurs at the feature level followed by a single classifier applied over the whole feature vector, and in the latter the pair of classifiers with max-relevance and min-redundancy is fused by means of a set of trainable methods and nontrainable rules to improve the final classification stage.

Regarding the experimental results, the following conclusions can be highlighted:

1. Trainable fusion methods: These fusion techniques got better results than the usual nontrainable rules, evidencing the feasibility of the proposed methods.
2. mRMR criteria: To prevent redundancy and to take advantage of the classifier diversity, avoiding empirical approaches, this method is very useful during the classifier selection, aiding in choosing the better combination in terms of mutual information.
3. LIDAR-based detection missing: The missing detections for the LIDAR-based classifiers are related mainly to the range distance to the object; the farther the object is from the LIDAR, the less range information is available and consequently the system is prone to false-negative classifications. Moreover, situations in which pedestrians appear very close to each other or close to or between other objects (especially cars) originate probable missing detections.
4. Vision-based detection missing: Most of the cases occur on low-contrast images and when the ROI background has intense texture.
5. Fusion strategy: The proposed fusion strategies achieved higher performance than the single classifiers, for which the decentralized scheme obtained the best result.

Moreover, in terms of practical applications, as the fusion schemes do not depend entirely on a single sensor space, this brings more robustness and safety to systems employing such detection schemes.

ACKNOWLEDGMENTS

This work is supported in part by the Fundação para a Ciência e a Tecnologia de Portugal (FCT)

under grant PTDC/EEA-ACR/72226/2006. C. Premebida is supported by the FCT under grant SFRH/BD/30288/2006 and O. Ludwig under grant SFRH/BD/44163/2008. The authors would like to thank the reviewers for their valuable comments and suggestions.

REFERENCES

Abe, S. (2005). Support vector machines for pattern classification (Advances in Pattern Recognition). Secaucus, NJ: Springer-Verlag New York, Inc.

Arras, K., Mozos, O., & Burgard, W. (2007, April). Using boosted features for the detection of people in 2D range data. In 2007 IEEE International Conference on Robotics and Automation, Rome (pp. 3402–3407).

Cheng, H., Zheng, N., Zhang, X., Qin, J., & van de Wetering, H. (2007). Interactive road situation analysis for driver assistance and safety warning systems: Framework and algorithms. IEEE Transactions on Intelligent Transportation Systems, 8(1), 157–167.

Dalal, N., & Triggs, B. (2005, June). Histograms of oriented gradients for human detection. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), Washington, DC (vol. 1, pp. 886–893). San Diego, CA: IEEE Computer Society.

Douillard, B., Fox, D., & Ramos, F. (2007, October). A spatio-temporal probabilistic model for multi-sensor object recognition. In IEEE/RSJ International Conference on Intelligent Robots and Systems, 2007. IROS 2007, San Diego, CA (pp. 2402–2408).

Gandhi, T., & Trivedi, M. M. (2007). Pedestrian protection systems: Issues, survey, and challenges. IEEE Transactions on Intelligent Transportation Systems, 8(3), 413–430.

Guivant, J. E., Masson, F. R., & Nebot, E. M. (2002). Simultaneous localization and map building using natural features and absolute information. Robotics and Autonomous Systems, 40(2–3), 79–90.

Hall, D., & Llinas, J. (1997). An introduction to multisensor data fusion. Proceedings of the IEEE, 85(1), 6–23.

Hwang, J. P., Cho, S. E., Ryu, K. J., Park, S., & Kim, E. (2007, September). Multi-classifier based lidar and camera fusion. In Intelligent Transportation Systems Conference, 2007. ITSC 2007, Seattle, WA (pp. 467–472). IEEE.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision, 60(2), 91–110.

Ludwig, O., Delgado, D., Gonçalves, V., & Nunes, U. (2009, October). Trainable classifier-fusion schemes: An application to pedestrian detection. In Intelligent Transportation Systems Conference, ITSC 2009, St. Louis, MO. IEEE.

Ludwig, O., & Nunes, U. (2008, October). Improving the generalization properties of neural networks: An application to vehicle detection. In 11th International IEEE Conference on Intelligent Transportation Systems, 2008. ITSC 2008, Beijing (pp. 310–315).

Mahlisch, M., Schweiger, R., Ritter, W., & Dietmayer, K. (2006, June). Sensorfusion using spatio-temporal aligned video and lidar for improved vehicle detection. In 2006 IEEE Intelligent Vehicles Symposium, Tokyo (pp. 424–429).

Pangop, L. N., Chapuis, R., Bonnet, S., Cornou, S., & Chausse, F. (2008, September). A Bayesian multisensor fusion approach integrating correlated data applied to a real-time pedestrian detection system. In IEEE IROS2008 2nd Workshop on Perception, Planning and Navigation for Intelligent Vehicles, Nice, France.

Peng, H., Long, F., & Ding, C. (2005). Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. IEEE Transactions on Pattern Analysis and Machine Intelligence, 27(8), 1226–1238.

Premebida, C., & Nunes, U. (2005, April). Segmentation and geometric primitives extraction from 2D laser range data for mobile robot applications. In Proceedings of 5th National Festival of Robotics, Scientific Meeting (ROBOTICA), Coimbra, Portugal.

Premebida, C., & Nunes, U. (2006, September). A multi-target tracking and gmm-classifier for intelligent vehicles. In Intelligent Transportation Systems Conference, 2006. ITSC '06, Toronto, Canada (pp. 313–318). IEEE.

Spinello, L., & Siegwart, R. (2008, May). Human detection using multimodal and multidimensional features. In IEEE International Conference on Robotics and Automation, 2008. ICRA 2008, Pasadena, CA (pp. 3264–3269).

Streller, D., & Dietmayer, K. (2004, June). Object tracking and classification using a multiple hypothesis approach. In 2004 IEEE Intelligent Vehicles Symposium, Parma, Italy (pp. 808–812).

Szarvas, M., Sakai, U., & Ogata, J. (2006, June). Real-time pedestrian detection using lidar and convolutional neural networks. In 2006 IEEE Intelligent Vehicles Symposium, Tokyo (pp. 213–218).

Tuzel, O., Porikli, F., & Meer, P. (2006, May). Region covariance: A fast descriptor for detection and classification. In Proceedings of 9th European Conference on Computer Vision, Graz, Austria (pp. 589–600).

Tuzel, O., Porikli, F., & Meer, P. (2007, June). Human detection via classification on Riemannian manifolds. In IEEE Conference on Computer Vision and Pattern Recognition, 2007. CVPR '07, Minneapolis, MN (pp. 1–8).

Vapnik, V. N. (1998). Statistical learning theory. New York: Wiley.

Xavier, J., Pacheco, M., Castro, D., Ruano, A., & Nunes, U. (2005, April). Fast line, arc/circle and leg detection from laser scan data in a player driver. In Proceedings of the 2005 IEEE International Conference on Robotics and Automation, 2005. ICRA 2005, Barcelona, Spain (pp. 3930–3935).

Zhang, Q., & Pless, R. (2004, September). Extrinsic calibration of a camera and laser range finder (improves camera calibration). In Proceedings. 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2004. (IROS 2004), Sendai, Japan (vol. 3, pp. 2301–2306).