Multiclass Multimodal Detection and Tracking in Urban Environments

Luciano Spinello[†, §] Rudolph Triebel[§] Roland Siegwart[§]

[†]Social Robotics Lab, University of Freiburg, Germany

[§]Autonomous Systems Lab, ETH Zurich, Switzerland

Abstract

This paper presents a novel approach to detect and track people and cars based on the combined information retrieved from a camera and a laser range scanner. Laser data points are classified by using boosted Conditional Random Fields (CRF), while the image based detector uses an extension of the Implicit Shape Model (ISM), which learns a codebook of local descriptors from a set of hand-labeled images and uses them to vote for centers of detected objects. Our extensions to ISM include the learning of object parts and template masks to obtain more distinctive votes for the particular object classes. The detections from both sensors are then fused and the objects are tracked using a Kalman Filter with multiple motion models. Experiments conducted in real-world urban scenarios demonstrate the effectiveness of our approach.

1 Introduction

One research area that has turned more and more into the focus of interest during the last years is the development of driver assistant systems and (semi-)autonomous cars. In particular, such systems are designed to operate in highly unstructured and dynamic environments. Especially in city centers, where many different kinds of transportation systems are encountered (walking, cycling, driving, etc.), the requirements for an autonomous system are very high. One key prerequisite is a reliable detection and distinction of dynamic objects, as well as an accurate estimation of their motion direction and speed. In this paper, we address this problem by focusing on the detection and tracking of people and cars. Our system is a robotic car equipped with cameras and a 2D laser range scanner. As we will show, the use of different sensor modalities helps to improve the detection results.

The system we present employs a variety of different methods from machine learning and computer vision, which have been shown to provide robust performances. We extend these methods obtaining substantial improvements and combine them into a complete system of detection, sensor fusion and object tracking. We use supervised learning techniques for both kinds of sensor modalities, which extract relevant information from large hand-labeled training data sets. In particular, the major contributions of this work are:

- Several extensions to the vision based object detector of Leibe et al. [2005], that uses a feature based voting scheme denoted as Implicit Shape Models (ISM). Our major improvements to ISM are the subdivision of objects into parts to obtain a more differentiated voting, the use of *template masks* to discard unlikely votes, and the definition of *superfeatures* that exhibit a higher evidence of an object's occurrence and are more likely to be found.
- The application and combination of boosted Conditional Random Fields (CRF) for classifying laser scans with the ISM based detector using vision. We use a Kalman Filter (KF) with multiple motion models to fuse the sensor information and to track the objects in the scene.

This paper is organized as follows. The next section describes work that is related to ours. Sec. 3 gives a brief overview of our overall object detection and tracking system. In Sec. 4, we introduce the implicit shape model (ISM) and present our extensions. Sec. 5 describes our classification method of 2D laser range scans based on boosted Conditional Random Fields. Then, in Sec. 6 we explain our sensor fusion techniques and our KF-based object tracker. Finally, we present experiments in Sec. 7 and conclude the paper.

2 Related Work

This section presents the scientific literature related to people and vehicle detection. It is organized in three parts: the first discusses range-based approaches, the second image-based methods, and the last one presents the related work in the area of multimodal detection using camera and laser range data.

2.1 Range-based methods

Several approaches can be found in the literature to identify a person in 2D laser data. A popular approach is to extract legs by detecting moving blobs that appear as local minima in the range data [Fod et al., 2002, Scheutz et al., 2004, Schulz et al., 2003]. They characterize people by computing geometrical and motion features. When motion features are used, people that do not move can not be detected. The work of Topp and Christensen [2005] overcomes this problem, obtaining good results in an cluttered indoor environment. Hähnel et al. [2003] consider the problem of classifying beams in range scans that are reflected by dynamic objects. An expectation maximization (EM) estimation is run in order to determine which beam has been reflected by a dynamic object as a person. The work of Xavier et al. [2005] is also based on the identification of people by geometrical features on the range scan. The data is segmented into clusters and a set of heuristics is applied in order to distinguish between lines, circles and legs. The first work that formulates the problem of detecting people as a learning problem in a principled manner has been developed by Arras et al. [2007]. Here, the authors use geometrical and statistical features extracted from clusters of the range scan to learn an AdaBoost classifier. Excellent results have been presented for indoor environments. Also, Luber et al. [2008] make use of learning techniques for detecting and tracking several classes of objects using unsupervised creation of exemplar models. In a later work, Arras et al. [2008] use a multi-hypotheses tracker to adaptively address the problem of occlusions and self-occlusions when tracking multiple pedestrians in range data. More recently, Lau et al. [2009] present an approach to track groups of people using distance clustering and a multi-hypotheses tracking system.

Detection of people in 3D range data is recently gaining attention in the robotics community. Navarro-Serment et al. [2009] use a ground detector, PCA analysis and geometrical descriptors classified by Support Vector Machines for detecting people from 3D data retrieved from several nodding laser rangefinders. In an own work [Spinello et al., 2010], we detect people in 3D point cloud data using a part-based voting approach with banks of trained AdaBoost classifiers. This method is more general as it does not need any ground detector, and yields very accurate detection results.

A very successful work in the field of vehicles detection using range data is the one of Petrovskaya and Thrun [2008], focussing on the tracking and detection of multiple vehicles via a model-based approach. It encompasses both geometric and dynamic properties of the tracked vehicle in a single Bayes filter. Other approaches based on segmentation and classification are the one of Zhao and Thorpe [1998] and Streller et al. [2002]. The first enforces a rectangular model of a car in range data by using heuristics on extracted lines and uses an Extended Interactive Motion Model for tracking. In the latter, several motion models are used and applied to simple geometrical models of vehicles.

2.2 Camera-based methods

In the area of image-based object detection, and people detection in particular, there mainly exist two kinds of approaches (see Enzweiler and Gavrila [2009] for a survey). One uses the analysis of a *detection window* or *templates* [Gavrila and Philomin, 1999, Viola et al., 2003], the other performs a *parts-based* detection [Felzenszwalb and Huttenlocher, 2000, Ioffe and Forsyth, 2001]. The detection window approach uses a scalable window that is scrolled through the image. For each step, a classification of the image area under the detection window is obtained. A template-based detection technique is similar to the previously described approach, but in this case a simple distance measure is computed between the edges in the image under the template silhouette and the silhouette itself. Part-based detection methods aim at independently detecting parts to obtain location hypotheses for entire objects. There exist plenty of computer vision based people detection systems described in the literature. Here, we refer to the most successful ones. Leibe et al. [2005] presented an image-based people detector using *Implicit Shape Models* (ISM) with excellent detection results in crowded scenes. This method is based on a database or *bag of words*, called codebook, extracted from standard descriptors, that vote for object centers. A mean shift mode estimation is used to define object hypotheses in the continuous space and a minimum description length method to select the winning ones. In earlier works, we showed already extensions of this method with a better feature selection and an improved nearest neighbor search [Spinello et al., 2008a,b]. Another image-based person detection algorithm that obtained remarkable detection results has been presented by Dalal and Triggs [2005]. This method is based on the classification of special image descriptors called Histogram of Oriented Gradients (HOG), computed over blocks of different sizes and scales in a fixed size detection window. The HOG descriptor is based on a collection of normalized image gradients on each cell. The resulting high dimensional vector is then classified with a linear support vector machine (SVM). Zhu et al. [2006] then refined this detector by using a fast rejector-based SVM cascade to discard the presence of a person in the detection window.

Unlike human bodies, cars have relatively uniform characteristics in structure such as four wheels, a certain number of pillars, two bumpers, etc. The appearance of these parts changes due to different car models, view points and lighting conditions. The methods already discussed for people detection are also used for detecting cars. Leibe et al. [2007] detect and track people and cars using a stereo system and an ISM approach where detection hypotheses are selected via an optimization that takes into account overlaps between detections and between object categories. Zheng and Liang [2009] compute ’strip features’ to describe image locations with arcs, edge-like and ridge-like patterns that are frequently found on vehicles. They learn a complexity-aware RealBoost to produce a fast and accurate classification method. Papageorgiou and Poggio [2000] detect cars and people by using an overcomplete set of Haar features classified with a support vector machine method.

2.3 Multimodal approaches

Most existing people detection methods based on camera *and* laser range data depend on hard constraints or on hand-tuned thresholding. Cui et al. [2005] use multiple laser scanners at foot height and a monocular camera to obtain people tracking by extracting feet and step candidates. Zivkovic and Kröse [2007] employ a range-based leg detector and boosted Haar features from camera images to detect people by using a probabilistic ruleset. Both methods cluster laser data points using a Canny edge detector and they extract unrobust image features to detect body parts. These approaches, based on simplistic processing of data, are hardly suited for outdoor scenarios due to the presence of clutter in image and range data. Moreover, in such

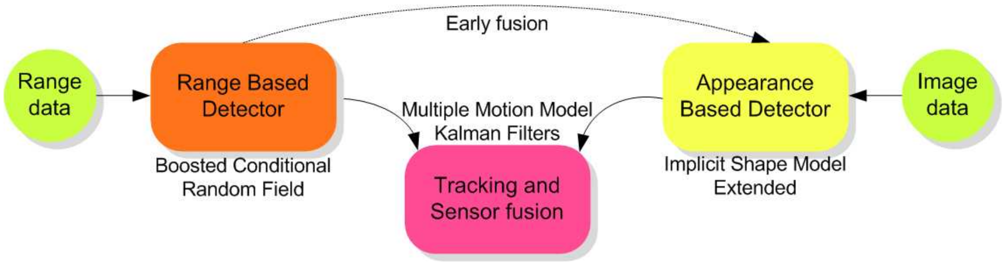
Early fusion
Range data
Range Based Detector
Boosted Conditional Random Field
Multiple Motion Model Kalman Filters
Tracking and Sensor fusion
Appearance Based Detector
Implicit Shape Model Extended
Image data

Figure 1: Overview of the method.

environments a large illumination variability can affect the descriptiveness of features that are based on simple intensity-based descriptors such as Haar features. The work of Schulz [2006] uses probabilistic exemplar models learned from camera and laser data and it applies a Rao-Blackwellized particle filter (RBPF) to track a person's appearance in the data. The RBPF tracks contours in the image based on Chamfer matching as well as point clusters in the laser scan and computes the likelihood of different prototypical shapes in the data. However, in outdoor urban scenarios occlusions are very likely, thus a contour matching approach is not an appropriate choice for dealing with partial object visibility. Neverthless, RBPF is a computationally demanding technique, especially when tracking multiple objects in a scene. Douillard et al. [2008] employ a Conditional Random Field (CRF) learned on 2D laser data and robust image features to detect multiple classes of objects (i.e. cars, people, vegetation). Promising results are obtained, but occlusions and overlapping object detection hypothesis, critical for yielding good results in any frame, are not handled by the algorithm. The work of Premebida et al. [2009] does not implement tracking of objects but it evaluates several centralized and decentralized fusion rules with standard vision and laser detectors. Wender and Dietmayer [2008] employ a camera and a laser scanner to detect cars in front of a robotic platform. They use simplistic heuristic rules on range data for estimating the viewpoint of the vehicle (front, side etc). Thus, they apply an AdaBoost-based image detector trained with Haar features on different car viewpoints.

3 Overview of Our Method

Our system consists of three main components (see scheme in Fig. 1):

- an appearance-based detector that processes data from a camera image
- a range based detector that processes data from a laser rangefinder
- a tracking module that fuses the information from both sensor modalities and provides an estimate of the motion vector for each tracked object.

The laser-based detector is based on a Conditional Random Field (CRF), formulated with a boosted set of geometrical and statistical features of 2D laser range data. The image based detector extends the multiclass version of the Implicit Shape Model (ISM) of Leibe et al. [2007]. The vision-based detector operates only on regions of interest obtained by projecting range data into the image to constrain the position and scale of the detectable objects (the "early fusion" step). The tracking module applies a Kalman Filter with two different motion models, fusing the information from camera and laser. In the following, we describe the particular components in detail.

Mathematical notations Throughout this paper we use the following mathematical notations:

- a *vector* is denoted with a bold letter, e.g. **a**.
- a *matrix* is denoted with a bold capital letter, for example **B**.
- *sets* are denoted with calligraphic capital letters, for example $\mathcal{C}$. The cardinality of a set $\mathcal{C}$ is expressed by the notation $\|\mathcal{C}\|$.
- numerical constants are denoted with capital letters.

4 Appearance Based Detection

Our vision-based people detector is mostly inspired by the work of Leibe et al. [2005] on scale-invariant Implicit Shape Models (ISM). In summary, an ISM consists in a set $\mathcal{I}$ of local region descriptors called *codebook*, and a set $\mathcal{V}$ of displacements and scale factors, usually named *votes*, for each descriptor. The idea is that each descriptor can be found at different positions inside an object and at different scales. Thus, a vote points from the position of the descriptor to the center of the object as it was found in the training data. To obtain an ISM from labeled training data, the descriptors are computed at interest point positions and then clustered, usually using agglomerative clustering with a maximal distance threshold ϑ_d. Then, the votes are obtained by computing the scale and the displacement of the objects' center to the descriptors. A training dataset consists in a collection of images and binary image masks defining the area and the position of the objects in each image. For the detection, new descriptors are computed on a test image and matched against the descriptors in the codebook. The votes that are cast by each matched descriptor are collected in a 3D *voting space*, and a maximum density estimator is used to find the most likely position and scale of an object.

In previous works, we presented already several improvements of the standard ISM approach [Spinello et al., 2008b,a]. Here, we show some more extensions of ISM to further improve the classification results. These extensions concern both the learning and the detection phase and are described in the following.

4.1 ISM Extension: Generating a Superfeature Codebook

In the standard ISM formulation, the process of generating a codebook does not include any feature selection. This has two potential disadvantages: first, a codebook for a given object category may contain many entries, and second, each entry may cast a big quantity of votes. One possibility to reduce the number of codebook entries is to increase the distance threshold ϑ_d when creating the codebook. However, in this case each entry in the codebook represents a larger variability of descriptors which leads to more votes per entry. When matching a codebook to new descriptors found in a test image, usually the same distance threshold ϑ_d is used as when generating the codebook. Therefore, if ϑ_d is large, more matches are found for a given new descriptor. Both effects result in a larger number of votes, which increases the number of false positive detections.

The goal of a *superfeature* codebook is to overcome these disadvantages by collecting more informed descriptors that cast stronger votes. We define superfeatures as features that are stable in image space and in descriptor space. This means that a superfeature is frequently found in the training set, at approximately the same image position with respect to the object center, and its variability in descriptor space is low. This definition ensures that for superfeatures a high evidence of the occurrence of the object is combined with a high probability to encounter an interest point. Let $\mathcal{O}^+$ be defined as the set of all interest points found inside the segmentation masks in the training data. Each element of $\mathcal{O}^+$ is a three-dimensional vector, where the dimensions are the relative displacement $(\Delta x, \Delta y)$ between the location of the interest point and the object center, and the scale s at which the interest point has been detected. Let furthermore κ be a function that maps from $\mathcal{O}^+$ to the D-dimensional descriptor space $\mathbb{R}^D$. In the training phase, κ is computed for all interest points in the labeled images. To compute superfeatures, we perform four steps. First, we determine points that lie in very dense areas of $\mathcal{O}^+$ by applying mean-shift mode estimation [Comaniciu et al., 2001]. This way, we obtain a reduced set $\mathcal{O}^*$ of interest points, i.e.:

$$\mathcal{O}^* = \text{ms}\Big(\rho_x, \rho_y, \rho_s, \mathcal{O}^+\Big), \tag{1}$$

where ms$(\cdot)$ indicates the mean shift estimator with uniform ellipsoidal kernel $\mathcal{K}$ of semiaxes ρ_x, ρ_y and ρ_s. We set $\rho_x = \rho_y$ in order to give equal importance to interest points found in both directions. In our implementation we use $\rho_x = \rho_y = 5$ and $\rho_s = 0.2$. Thus, $\mathcal{O}^*$ consists of the M *modes* $\mathbf{o}_1^*, \ldots, \mathbf{o}_M^*$ of the interest point distribution in $\mathcal{O}^+$ as found by the mean-shift estimator.

In the second step, we determine for each mode $\mathbf{o}_i^*$ the set $\mathcal{J}_i$ of image descriptors that have been computed at interest points inside the kernel around $\mathbf{o}_i^*$, i.e.

$$\mathcal{J}_i = \Big\{\kappa(\mathbf{p}) \mid \mathbf{p} \in \mathcal{O}^+ \cap \mathcal{K}(\mathbf{o}_i^*)\Big\}, \tag{2}$$

where $\mathcal{K}(\mathbf{o}_i^*)$ denotes the ellipsoidal kernel centered at the mode $\mathbf{o}_i^*$. Then, we apply agglomerative clustering with average linkage to the descriptors in $\mathcal{J}_i$, i.e.

$$\{C_1, C_2, \ldots\} = \mathrm{ac}\,(\vartheta_d, \mathcal{J}_i)\,, \tag{3}$$

where ac$\,(\cdot)$ represents a function that computes agglomerative clustering with distance threshold ϑ_d, and $C_1, C_2, \ldots$ are the resulting clusters in descriptor space.

In the last step, we remove all clusters with cardinality smaller than a threshold ϑ_c and store the centroids of those clusters that are bigger than the median of the cardinality of the remaining clusters into the descriptor set $\mathcal{I}_i^*$, or formally

$$\mathcal{I}_i^* := \{\mathrm{cn}(C) \mid C \in \mathfrak{C} \;\wedge\; \|C\| \geq \mathrm{md}(\mathfrak{C})\}\,. \tag{4}$$

Here, $\mathfrak{C}$ denotes the set of all clusters that are bigger than ϑ_c, cn$(\cdot)$ computes the centroid of a cluster, and md$(\cdot)$ returns the median cluster cardinality. The resulting superfeature codebook $\mathcal{I}^*$ is defined as

$$\mathcal{I}^* := \bigcup_{i=1}^{M} \mathcal{I}_i^*\,. \tag{5}$$

The computation of the set of votes $\mathcal{V}^*$ for $\mathcal{I}^*$ follows the same procedure as in standard ISM.

The resulting superfeature codebook $\mathcal{I}^*$ has less elements than the standard ISM codebook and each entry is associated to less votes. Figure 2 shows a visual explanation of the superfeature codebook generation. It is interesting to see that the superfeatures inherently reflect the skeleton of the object. In case of a pedestrian, superfeatures are mostly taken in the Λ-shaped area between the legs, and nearby the shoulders. Even though this result is strictly related to the kind of interest point detector (e.g. Harris and Hessian interest points are located either on corners or on blobs), it intuitively reflects distinctive local areas for detecting pedestrians. This result is in agreement with other local weighting methods found in the area of image-based people detection (see e.g. the discussion of Dalal and Triggs [2005] on the high classification weight that such areas receive).

4.2 ISM Extension: Learning Object Parts

The aim of this procedure is to further enrich the information retrieved in the voting process by distinguishing between different object parts from which the vote has been cast. The segmentation into parts is computed offline during the training process for each object category. Here, an object part is defined as a sector of a circle, where the circle center is aligned with the center of the bounding box that encompasses all training instances of an object class. This definition of an object part is motivated by the fact that the displacement vectors stored in an ISM vote for object centers. Hence, a natural way to distinguish the voters in the training data is with respect to the orientation of their displacement vectors.

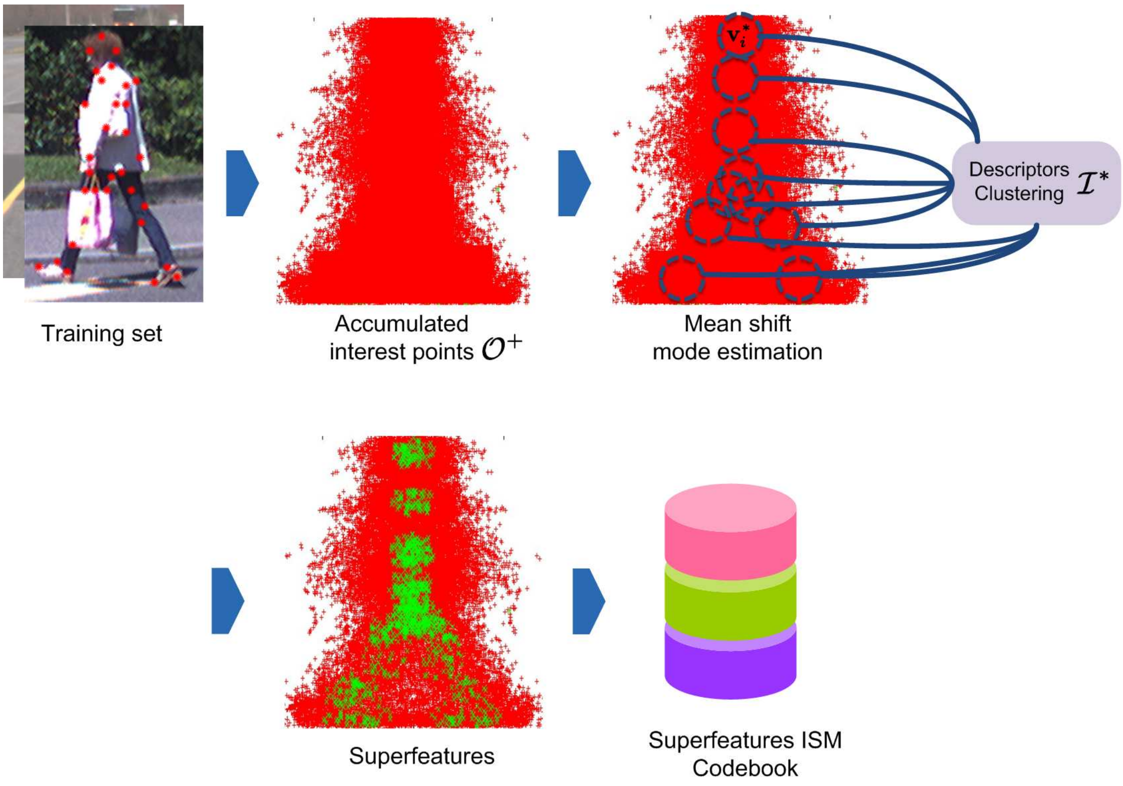
Training set
Accumulated interest points $\mathcal{O}^+$
Mean shift mode estimation
$\mathbf{v}_i^*$
Descriptors Clustering $\mathcal{I}^*$
Superfeatures
Superfeatures ISM Codebook

Figure 2: Generation of a superfeature codebook. Superfeatures are stable features in image and descriptor space. First, all interest points from the training data are accumulated in a continuous space. Then, high density areas are found using mean shift mode estimation. In the next step, we consider the descriptors associated to the clustered points and segment them using agglomerative clustering. From the resulting clusters, we select those that are larger than the median and store them together with the votes from the associated interest points in the superfeature codebook. In this example, we used Shape Context descriptors computed at Hessian interest points (in red) for the class 'pedestrian'. The position of the superfeatures are depicted in green.

To distinguish appropriate object parts, we perform three steps. We start again with the accumulated set $\mathcal{O}^+$ of interest points from the training data set. Then, we compute the orientation angle of each displacement vector with respect to the horizontal line through the center of the bounding box that encompasses all object instances (see Fig. 3). All orientation angles are collected in a set $\mathcal{A}$. Finally, we apply k-means clustering [Lloyd, 1982] to the elements of $\mathcal{A}$. The problem here is that the number K of clusters is not given beforehand. We solve this by re-running the clustering algorithm with increasing values of K and evaluating the resulting clusters with the Bayesian Information Criterion (BIC) [Schwarz, 1978]. The BIC can be used

Algorithm 1: K-means clustering with estimation of the number of clusters.

```
Input: Set of orientation angles $\mathcal{A}$ from voters
Output: Optimal set of clusters $\mathfrak{A}^*$
$K \leftarrow 1$
$b_{old} \leftarrow -\infty$
$b_{new} \leftarrow -\infty$
$\mathfrak{A} \leftarrow \emptyset$
while $b_{new} \geq b_{old}$ do
    $\mathfrak{A}^* \leftarrow \mathfrak{A}$
    $\mathfrak{A} \leftarrow$ kMeans $(\mathcal{A}, K)$
    $b_{old} \leftarrow b_{new}$
    $b_{new} \leftarrow -2\ln(\frac{\mathtt{RSS}(\mathfrak{A})}{||\mathcal{A}||}) + K\ln(||\mathcal{A}||)$      Compute BIC using residual sum of squares (RSS)
    $K \leftarrow K + 1$
end
return $\mathfrak{A}^*$
```

for model selection from a class of parametric models with different numbers of parameters. It represents a balanced score based on the likelihood of the model and its complexity. Our overall clustering method is summarized in Algorithm 1. We note that the Residual Sum of Squares (RSS) of clusters obtained with the k-means algorithm decreases monotonically with growing K. The RSS is exactly 0 when $K = ||\mathcal{A}||$, i.e. when each data point defines its own cluster. The BIC is used to trade off a low residual error with a low model cost. Once the BIC does not increase any longer, the maximum is found and the process stops. To perform k-means clustering on $\mathcal{A}$, we need to take care of the fact that the orientation angles are periodic, i.e. 0 needs to be identified with 2π. Fortunately, in k-means clustering only relative distances between points and clusters are required. Thus, we can replace each element in $\mathcal{A}$ by a corresponding point on the unit circle and use the arc length between two such points as the distance metric for clustering. When clustering is completed, $\mathcal{A}$ is represented by a collection of angle intervals: $\mathcal{A} = (a_1, \ldots, a_K]$, where $a_i = [\alpha_{i-1}, \alpha_i)$ is an angle interval that defines an object part.

An example of the outcome of our clustering algorithm is shown in Fig. 4. Note that although our algorithm does not explicitly search for a semantical subdivision of the object (e.g.: legs, arms, etc. in case of the pedestrian object category), it nevertheless resembles this automatically without human interaction. In Sec. 4.4 we describe how we use this extended shape information for hypothesis selection.

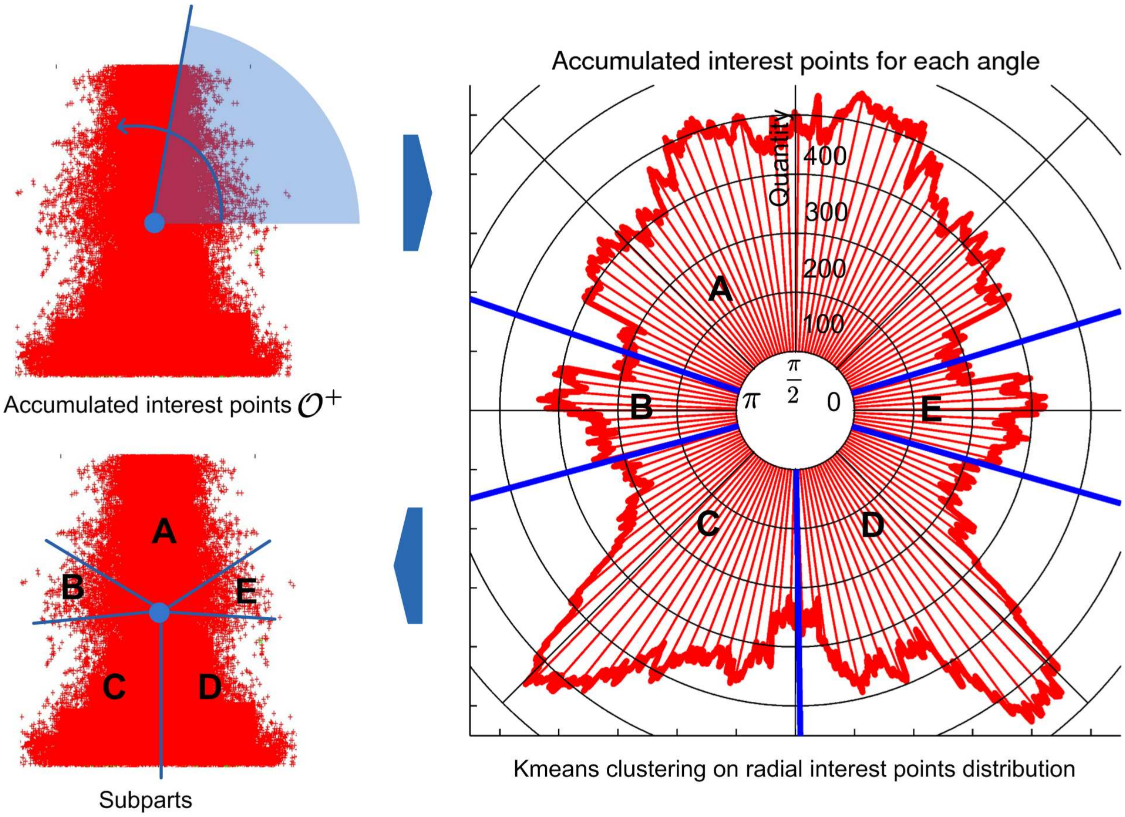

Figure 3: Subparts are computed by accumulating interest points, iteratively running k-means clustering, and using the BIC to score the cluster result.

4.3 ISM Extension: Learning Shape Templates

Based on a similar reasoning as described in the previous section, we propose another extension to the standard ISM approach to distinguish the votes with respect to their quality. The aim of this is to discard *outlier votes*, i.e. those that are cast from interest points located in unlikely areas for a given object class. Outliers are caused by training examples with an unusual shape where some interest points lie outside the most likely shape of the object. For example, there might be training examples of the class “pedestrian”, where a person extremely extends the arms. Then, if there are interest points detected on an arm, the resulting displacement vector stored in $\mathcal{V}$ will be very rare and thus correspond to an unlikely vote. Later, in the detection phase, this causes problems, because such an unlikely vote is treated in the same way as likely ones, causing many false positive detections.

A first attempt to detect and remove outlier votes has already been made by Leibe et al. [2005]. There the authors compute a combined optimization between expected segmentation

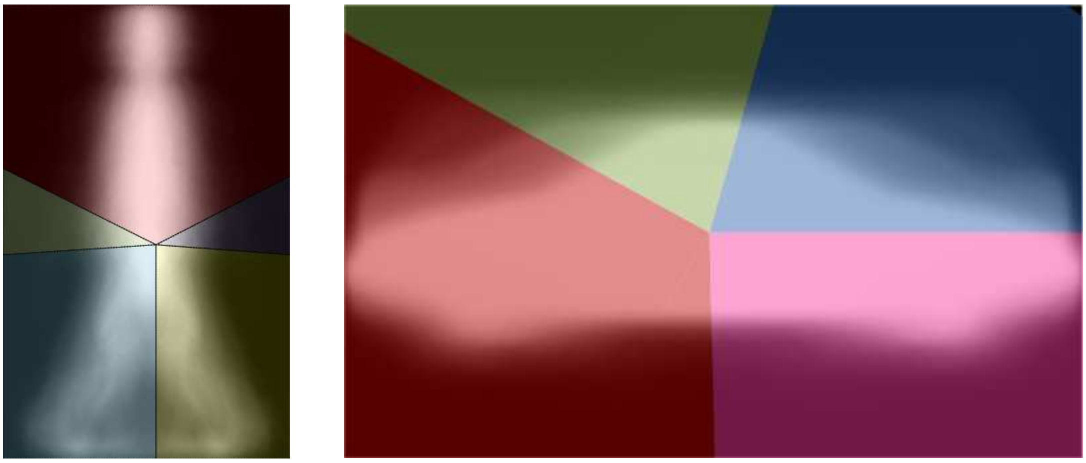

Figure 4: Clustered object parts (colored sectors) and template masks, overlaid as brightness values, for the classes *pedestrian* and *car-sideview*. Both are computed from the training set. Note that even though the object parts are computed unsupervised, they exhibit some semantic interpretation.

and silhouette matching via Chamfer matching [Borgefors, 1988]. This approach is computationally expensive and influenced by noise due to the nature of contour matching. In contrast, we propose a probabilistic approach. Instead of relying on the object's silhouette to determine outliers, we use the entire binary shape masks from the training data. By aligning all shape masks for a given object class so that their center points coincide and by computing the average mask, we obtain a gray value mask T_c with pixel values between 0 and 1. This procedure is similar to the one used to produce eigenfaces [Sirovich and Kirby, 1987]. These pixel values can be interpreted as prior probabilities for the location of interest points in the given object class. We denote T_c as the *template mask* of the object class c. Naturally, all training images used to create the template mask are given in scale 1, but we can obtain template masks at different scales by scaling T_c using bilinear interpolation.

An example of the template masks which we obtained for the classes "pedestrian" and "car-sideview" is shown in Fig. 4. Here, the template masks are visualized as brightness values together with the part clustering method presented in the previous section. As can be seen, the average shapes of both object classes are clearly visible.

4.4 ISM Extension: Multiclass Hypothesis selection

After learning standard codebooks $\mathcal{I}_c$, superfeature codebooks $\mathcal{I}^*_c$, segmented object parts $\mathcal{A}_c$, and template masks T_c for each object class $c = 1, \ldots, C$ from the training data as described before, we incorporate these information into the detection step. Here, we have to perform some further adaptation to the standard ISM approach, as we assume a multi-class problem.

Before however, we formulate the detection step mathematically.

After computing interest points and shape descriptors for a given test image, the latter are matched with all codebooks $\mathcal{I}_c$ and $\mathcal{I}_c^*$, and the modes of the voting space are computed using mean-shift, as described above. Let $\mathbf{h}_c = (\overline{x}_c, \overline{y}_c, s_c)$ be a resulting mode, i.e. a possible center location $(\overline{x}_c, \overline{y}_c)$ of an object of class c and its scale s_c. We will refer to $\mathbf{h}_c$ as a *hypothesis* of class c. Furthermore, let $\mathcal{X}_c$ be the interest point locations $\mathbf{x}_i$ of all voters that were responsible to create hypothesis $\mathbf{h}_c$. As in standard ISM, each vote has an assigned voting strength w_i. In the following, we will include the voting strength as an additional dimension to the point location vector, i.e. $\mathbf{x}_i = (x_i, y_i, s_i, w_i)$. Using this, we define a *voting score* as

$$\mathrm{vs}(\mathbf{h}_c) = \sum_{\mathbf{x}_i \in \mathcal{X}_c} 2^{b_i} w_i T_c(x_i, y_i, \mathbf{h}_c), \quad \text{where} \quad b_i = \begin{cases} 1 & \text{if } \mathbf{x}_i \text{ results from } \mathcal{I}_c^* \\ 0 & \text{if } \mathbf{x}_i \text{ results from } \mathcal{I}_c \end{cases} \tag{6}$$

and $T_c(x_i, y_i, \mathbf{h}_c)$ is the evaluation of the template mask at position (x_i, y_i) after placing its center at $(\overline{x}_c, \overline{y}_c)$ and rescaling it with s_c (see above). This means that the quality of a hypothesis is influenced by four values, namely the number of votes, their strength w_i, whether they arise from a superfeature match, and the prior quality of the voters obtained from the shape template T_c. Unlikely votes with respect to the shape template receive a very low weight and their contribution to the hypothesis score is strongly reduced.

Furthermore, for each object class c we make use of the information of the learned subparts $\mathcal{A}_c$. The idea is to obtain an information about the amount of parts that have been detected. Intuitively, a foreground object is expected to have most of the parts well detected, instead, an occluded object appears with less parts. To account for the different object parts from which votes may be cast, we first formulate the voting score vs_k, which is *restricted* to an interval $\mathbf{a}_k = (\alpha_{k-1}, \alpha_k)$ of orientations of vote vectors, where $k = 1, \ldots, K$ is the index of the corresponding object part, i.e.

$$\mathrm{vs}_k(\mathbf{h}_c) = \sum_{\mathbf{x}_i \in \mathcal{X}_c,\ \alpha_{k-1} \le \alpha(\mathbf{x}_i) < \alpha_k} w_i T_c(x_i, y_i, \mathbf{h}_c) \quad \text{and} \quad \alpha(\mathbf{x}_i) = \arctan\left(\frac{y_i - \overline{y}_c}{x_i - \overline{x}_c}\right) \tag{7}$$

All part-based scores are then collected in a K-dimensional vector ξ defined as

$$\xi(\mathbf{h}_c) = (\mathrm{vs}_1(\mathbf{h}_c), \ldots, \mathrm{vs}_K(\mathbf{h}_c)). \tag{8}$$

Intuitively, this is a weighted histogram of votes where each bin corresponds to a learned object part, or equally a sector of vote orientations.

To find the best hypothesis we define a partial order $\prec$ on all hypotheses based on a function Δ_r as

$$\mathbf{h}_i \prec \mathbf{h}_j \Leftrightarrow \Delta_r\left(\xi(\mathbf{h}_i), \xi(\mathbf{h}_j)\right) < 0 \quad \text{where} \quad \Delta_r\left(\xi(\mathbf{h}_i), \xi(\mathbf{h}_j)\right) := \sum_{k=1}^{K} \mathrm{sign}\left(\xi_k(\mathbf{h}_i) - \xi_k(\mathbf{h}_i)\right) \tag{9}$$

Algorithm 2: Multiclass detection with ISMe

```
Input:
  • Interest points $\mathbf{x}_i$ and corresponding shape descriptors $\mathbf{d}_i$ from a new test image
  • codebooks $\mathcal{I}_1,\dots,\mathcal{I}_C$, superfeature codebooks $\mathcal{I}^*_1,\dots,\mathcal{I}^*_C$ and votes $\mathcal{V}_1,\dots,\mathcal{V}_C$
    for all $C$ object classes
  • minimal hypothesis score $\sigma_{min}$
Output: Set of optimal object hypotheses $\mathcal{H}^*$
$\mathcal{H}^* \leftarrow \emptyset$
$h_{win} \leftarrow \infty$
while $h_{win} > \sigma_{min}$ do
    for $c = 1$ to $C$ do
        $\mathcal{D}_c \leftarrow$ FindMatches $(\mathcal{I}_c, \{\mathbf{d}_i\})$
        $\mathcal{D}^*_c \leftarrow$ FindMatches $(\mathcal{I}^*_c, \{\mathbf{d}_i\})$
        $\mathcal{Y}_c \leftarrow$ CollectVotes $(\mathcal{D}_c, \mathcal{D}^*_c, \mathcal{V}_c)$
        $\mathcal{H}_c \leftarrow \mathrm{ms}(\rho_x, \rho_y, \rho_s, \mathcal{Y}_c)$        Mean-shift operation, returns set of hypotheses
        Find $\mathbf{h}^*_c$ s.t. $\mathbf{h}_c \prec \mathbf{h}^*_c \;\forall \mathbf{h}_c \in \mathcal{H}_c, \mathbf{h}_c \neq \mathbf{h}^*_c$        Best hypothesis for class $c$, see Eqn. (9)
        $\Gamma_c \leftarrow$ ComputeHypothesisArea $(\mathbf{h}^*_c)$        see Eqn. (10)
    end
    $\mathbf{h}^* \leftarrow \mathrm{argmax}_{\mathbf{h}^*_c}(\Gamma_1,\dots,\Gamma_C)$
    $h_{win} \leftarrow \mathrm{vs}(\mathbf{h}^*)$
    $\mathcal{H}^* \leftarrow \mathcal{H}^* \cup \mathbf{h}^*$
end
return $\mathcal{H}^*$
```

where $\xi_k(\mathbf{h}_i)$ indicates the value contained in the bin k of the histogram for the hypothesis $\mathbf{h}_i$. Intuitively, the function Δ_r measures for which of the hypothetical objects the individual object parts are stronger represented in the voting space. Using Eqn. (9), we can determine the hypothesis $\mathbf{h}^*_c$ with the highest order of all hypotheses for class c. In case of ambiguity we use the one with the highest global score $\mathrm{vs}(\cdot)$.

However, to determine the strongest hypothesis across all object classes, we can not simply compare the scores, as they are based on different codebooks with different numbers of entries. Instead, we use another measure based on the object *area* that is covered by a hypothesis. The idea here is that all point locations in $\mathcal{X}_c$ of votes that were responsible for $\mathbf{h}_c$, can be viewed as small patches inside an object that contribute to the entire shape of the object, just as pieces of a puzzle. To formulate that, we define a square region $\gamma(\mathbf{x}_i)$ around each $\mathbf{x}_i$ with side length proportional to the scale s_i. For the hypothesis $\mathbf{h}_c$ we can then define the relative area covered

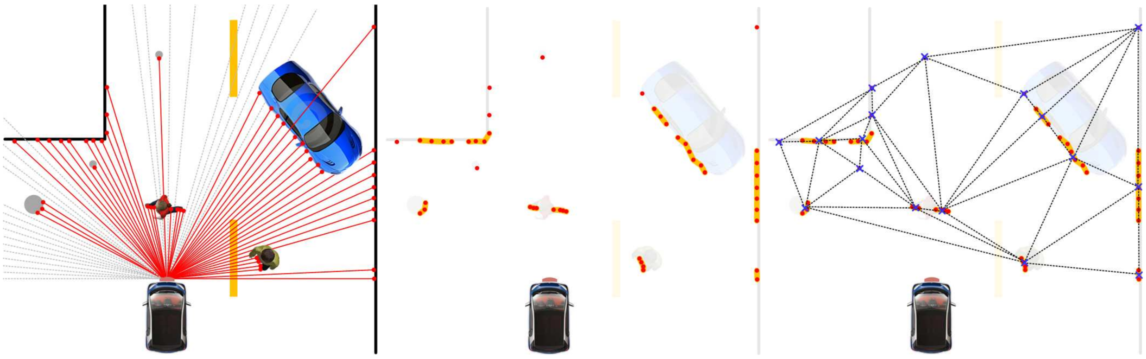

Figure 5: An urban environment with cars, pedestrian and other objects as it is perceived by a 2D laser. **Left**: Laser beams are shown in red, circles represents the measured points. Gray beams indicate out of range data due to material reflections, sun related effects and particular object poses. **Center**: Resulting JDC clustering of the scene. Orange lines depicts consecutive points segmented in the same cluster. **Right**: A Delaunay triangulation is build on the centroids of the segments. This defines a graph among segments.

by all vote patches as

$$\Gamma_c = \frac{\text{area}\left(\bigcup_{\mathbf{x}_i \in \mathcal{X}_c} \gamma(\mathbf{x}_i)\right)}{\|\{(x,y) \mid T(x,y,\mathbf{h}_c) \geq 0.5\}\|}, \tag{10}$$

where the function area(·) computes the area of the joint region, and the denominator approximates the area of the object by counting all points in the shape template that are likely to be inside the shape. Care has to be taken in the case of overlapping class hypotheses. Here, we compute the set intersection of the interest points in the overlapping area and assign their corresponding γ values alternately to one and the other hypothesis.

Once an optimal hypothesis $\mathbf{h}^*$ across all classes is found, we remove all the votes coming from those features that contributed to $\mathbf{h}^*$, because we assume that an image feature belongs to just a single object. The scores are then recomputed until a minimum score σ_{min} is reached. Algorithm 2 summarizes the individual steps.

5 Structure Based Detection

For the detection of objects in 2D laser range scans, several approaches have been presented in the past [see for example Arras et al., 2007]. Most of them have the disadvantage that they disregard the conditional dependence between data in a close neighborhood. In particular, they can not model the fact that the label l_i of a given laser segment $\mathcal{S}_i$ is more likely to be l_j if we know that l_j is the label of $\mathcal{S}_j$ given that $\mathcal{S}_j$ and $\mathcal{S}_i$ are neighbors. One way to model this conditional dependency is to use Conditional Random Fields (CRFs) [Lafferty et al., 2001], as shown by Douillard et al. [2008]. CRFs represent the conditional probability $p(\mathbf{l} \mid \mathbf{s})$ using an undirected cyclic graph, in which each node is associated with a hidden random variable l_i and an observation $\mathbf{s}_i$. In our case, l_i is a discrete label that ranges over 3 different classes (pedestrian, car and background) and $\mathbf{s}_i$ is a feature vector extracted from the 2D segment $\mathcal{S}_i$ in the laser scan. A preprocessing step on range data has been defined in order to produce segments for the CRF detector. We use a simple clustering technique to group nearby points, called Jump Distance Clustering (JDC). It is fast and simple to implement: if the Euclidean distance between two adjacent data points exceeds a given threshold, a new cluster is generated otherwise the point is added to the current cluster (see Fig. 5-center). Each cluster, or segment, is defined as the set of points $\mathcal{S}_i$. Moreover we compute a Delaunay triangulation between the centroids of each segment $\mathcal{S}_i$ in order to create a graph that connects clusters, see Fig. 5-right.

Assuming a maximal clique size of 2 for the graph, we can compute the conditional probability of the labels $\mathbf{l}$ given the observations $\mathbf{s}$ as:

$$p(\mathbf{l} \mid \mathbf{s}) = \frac{1}{Z(\mathbf{s})} \prod_{i=1}^{N} \varphi(\mathbf{s}_i, l_i) \prod_{(i,j)\in\mathcal{E}} \psi(\mathbf{s}_i, \mathbf{s}_j, l_i, l_j), \tag{11}$$

where $Z(\mathbf{s}) = \sum_{\mathbf{l}'} \prod_{i=1}^{N} \varphi(\mathbf{s}_i, l'_i) \prod_{(ij)\in\mathcal{E}} \psi(\mathbf{s}_i, \mathbf{s}_j, l'_i, l'_j)$ is usually called the *partition function*, $\mathcal{E}$ is the set of edges in the graph, and φ and ψ represent node and edge potentials. To determine φ and ψ we use the log-linear model

$$\begin{aligned} \varphi(\mathbf{s}_i, l_i) &= e^{\mathbf{u}_n \cdot \mathbf{f}_n(\mathbf{s}_i, l_i)} && (12) \\ \psi(\mathbf{s}_i, \mathbf{s}_j, l_i, l_j) &= e^{\mathbf{u}_e \cdot \mathbf{f}_e(\mathbf{s}_i, \mathbf{s}_j, l_i, l_j)}, && (13) \end{aligned}$$

where $\mathbf{f}_n$ and $\mathbf{f}_e$ are feature functions for the nodes and the edges in the graph, and $\mathbf{u}_n$ and $\mathbf{u}_e$ are feature weights that are determined in the training phase. The computation of the partition function Z is intractable due to the exponential number of possible labelings $\mathbf{l}'$. Instead, we compute the *pseudo-likelihood*, which approximates $p(\mathbf{l} \mid \mathbf{s})$ and is defined by the product of all likelihoods computed on the *markov blanket* (direct neighbors) of node i, i.e.

$$p(\mathbf{l} \mid \mathbf{s}) \approx pl(\mathbf{l} \mid \mathbf{s}) = \prod_{i=1}^{N} \frac{\varphi(\mathbf{s}_i, l_i) \prod_{\mathbf{s}_j \in \mathcal{N}(\mathbf{s}_i)} \psi(\mathbf{s}_j, \mathbf{s}_i, l_j, l_i)}{\sum_{\mathbf{l}'} (\varphi(\mathbf{s}_i, l'_i) \prod_{\mathbf{s}_j \in \mathcal{N}(\mathbf{s}_i)} \psi(\mathbf{s}_j, \mathbf{s}_i, l'_i, l'_j))}. \tag{14}$$

Here, $\mathcal{N}(\mathbf{s}_i)$ denotes the set of direct neighbors of node i. In the training phase, we compute the weights $\mathbf{u} = (\mathbf{u}_n, \mathbf{u}_e)$ that minimize the negative log pseudo-likelihood together with a Gaussian shrinkage prior as proposed by Ramos et al. [2007]:

$$L(\mathbf{u}) = -\log pl(\mathbf{l} \mid \mathbf{s}) + \frac{(\mathbf{u} - \hat{\mathbf{u}})^T (\mathbf{u} - \hat{\mathbf{u}})}{2\sigma^2}. \tag{15}$$

For the minimization of L, we use the L-BFGS gradient descent method [Liu and Nocedal, 1989]. Once the weights are obtained, they are used in the inference phase to find the labels $\mathbf{l}$ that maximize Eqn. (11). Here, we do not need to compute the partition function Z, as it is not dependent on $\mathbf{l}$. We use max-product loopy belief propagation (BP) to find the distributions of each label l_i. The final labels are then obtained as those that are most likely for each node.

In our case the Delaunay triangulation among segments defines the structure of the network. We use a set of statistical and geometrical features for the nodes of the CRF, e.g. width, circularity, standard deviation, kurtosis, etc. [for a full list see Spinello and Siegwart, 2008]. However, we do not use these features directly in the CRF, because, as stated by Ramos et al. [2007] and also from our own observation, the CRF is not able to handle non-linear relations between the observations and the labels. Instead, we apply AdaBoost [Freund and Schapire, 1997] to the node features and use the outcome as features for the CRF. For our particular classification problem with multiple classes, we train one binary AdaBoost classifier for each class against the others. As a result, we obtain for each class c a set of M weak classifiers h_i^c (in this case decision stumps) and corresponding weight coefficients α_i^c so that the sum

$$g_c(\mathbf{s}_i) := \sum_{i=1}^{M} \alpha_i^c h_i^c(\mathbf{s}_i) \tag{16}$$

is positive for observations assigned with the class label c and negative otherwise. Note that the absolute value of g_c can be interpreted as a classification quality. To obtain a classification *likelihood*, we apply the logistic function $a(x) = (1+e^{-x})^{-1}$ to g_c. We do this for two reasons: first the resulting values are between 0 and 1 and can be interpreted as likelihoods of corresponding to class c. Second, by applying the same technique also for the edge features, the resulting potentials are better comparable. Thus, the node feature function $\mathbf{f}_n$ of the segment features $\mathbf{s}_i$ and the label l_i is computed as

$$\mathbf{f}_n(\mathbf{s}_i, l_i) = a(g_{l_i}(\mathbf{s}_i)). \tag{17}$$

For the edge features $\mathbf{f}_e$ we compute two values, namely the Euclidean distance between the centroids $\mathbf{c}_i$ and $\mathbf{c}_j$ of the segments $\mathcal{S}_i$ and $\mathcal{S}_j$, along with a value g_{ij} defined as

$$g_{ij}(\mathbf{s}_i, \mathbf{s}_j) = \text{sign}(g_i(\mathbf{s}_i)g_j(\mathbf{s}_j)) \cdot (|g_i(\mathbf{s}_i)| + |g_j(\mathbf{s}_j)|). \tag{18}$$

Thus, the value of g_{ij} has a positive sign if AdaBoost classifies $\mathbf{s}_i$ and $\mathbf{s}_j$ into the same class, and otherwise it is negative. The absolute value of g_{ij} is the sum of the classification qualities of AdaBoost. If $g_i(\mathbf{s}_i)$ and $g_j(\mathbf{s}_j)$ are far from 0 then g_{ij} has a high value, and viceversa. To summarize, we define the edge features as

$$\mathbf{f}_e(\mathbf{s}_i, \mathbf{s}_j, l_i, l_j) = \begin{cases} \begin{pmatrix} a(\|\mathbf{c}_i - \mathbf{c}_j\|) & a(g_{i,j}(\mathbf{s}_i, \mathbf{s}_j)) \end{pmatrix}^T & \text{if } l_i = l_j \\ \begin{pmatrix} 0 & 0 \end{pmatrix}^T & \text{otherwise.} \end{cases} \tag{19}$$

The intuition behind equation (19) is that edges that connect segments with equal labels have a non-zero feature value and thus yield a higher potential. The latter is sometimes referred to as the generalized Potts model [see Potts, 1952].

6 Object Tracking and Sensor Fusion

In this section we explain how to combine the two sensor modalities together. Range and image data is used for “early fusion” and then combined in the tracking system. The early fusion step consists in a technique to constrain the vision-based detector in salient image regions. The tracker combines detection results from camera and laser data and solves the data association.

An important factor in our multisensor system is the extrinsic calibration between camera and laser. The internal camera parameters are estimated using the camera calibration method by Zhang [1999]. Then, we employ the method explained by Pless and Zhang [2004] to calibrate the 2D laser rangefinder with the camera. The procedure consists in simultaneously collecting image and range data of a planar checkerboard placed in front of a robot at different positions and orientations. For each pose of the planar pattern, the method constrains the extrinsic parameters by registering the laser scanline on the planar pattern with the estimated plane computed from the image. The solution uses nonlinear optimization that minimizes the re-projection error.

6.1 Early fusion: using laser segments to bound the voting space

The early fusion method is concerned with the definition of constraints in the ISMe voting space of the image-based detector, in order to generate more precise object hypotheses. The idea is to project segments extracted from the laser data as 3D boxes in the voting space. If we consider a single laser segment, it could be projected as a box with a height set to a fixed value, a width defined by the extremal points of the segment $\mathcal{S}_i$, and a depth defined by the scale tolerance $\vartheta_{\mathcal{S}_i}$. These 3D boxes define boundaries in the voting space for hypothesis selection for the image detector. Before the image hypothesis selection is run, the early fusion takes place and removes hypotheses that are not compatible with the boundaries. The generous dimensions of the boxes allow the survival of imprecise detections in position and scale.

In order to consider range-images in the early fusion process, we need to set $\vartheta_{\mathcal{S}_i}$ for each object class. Precisely, we need to compute $\vartheta_{\mathcal{S}_i}$ as a function of the laser segment distance. We assume, for practical reasons, that the relationship between these two variables is linear, even though this is not true due to lens distorsions. The idea is to perform a linear least-squares regression that relates the objects’ pixel heights $\boldsymbol{\omega}^s$ with the object distances $\boldsymbol{\omega}^d$:

$$\omega_i^d = \beta_1 \omega_i^s + \beta_2, \tag{20}$$

where (β_1,β_2) are the parameters of the line computed with the regression from a collection of measured object heights and distances. Thus, we are able to compute a *hallucinated* distance for each object category from a given input scale (and viceversa).

The scale ω^s estimated for each segment distance is then converted into the depth of the 3D region of interest in the ISMe voting space in order to easily prune false image detection hypotheses:

$$\vartheta_{\mathcal{S}_i} = (\omega_i^s - \vartheta_{\mathcal{S}}^*, \omega_i^s + \vartheta_{\mathcal{S}}^*), \tag{21}$$

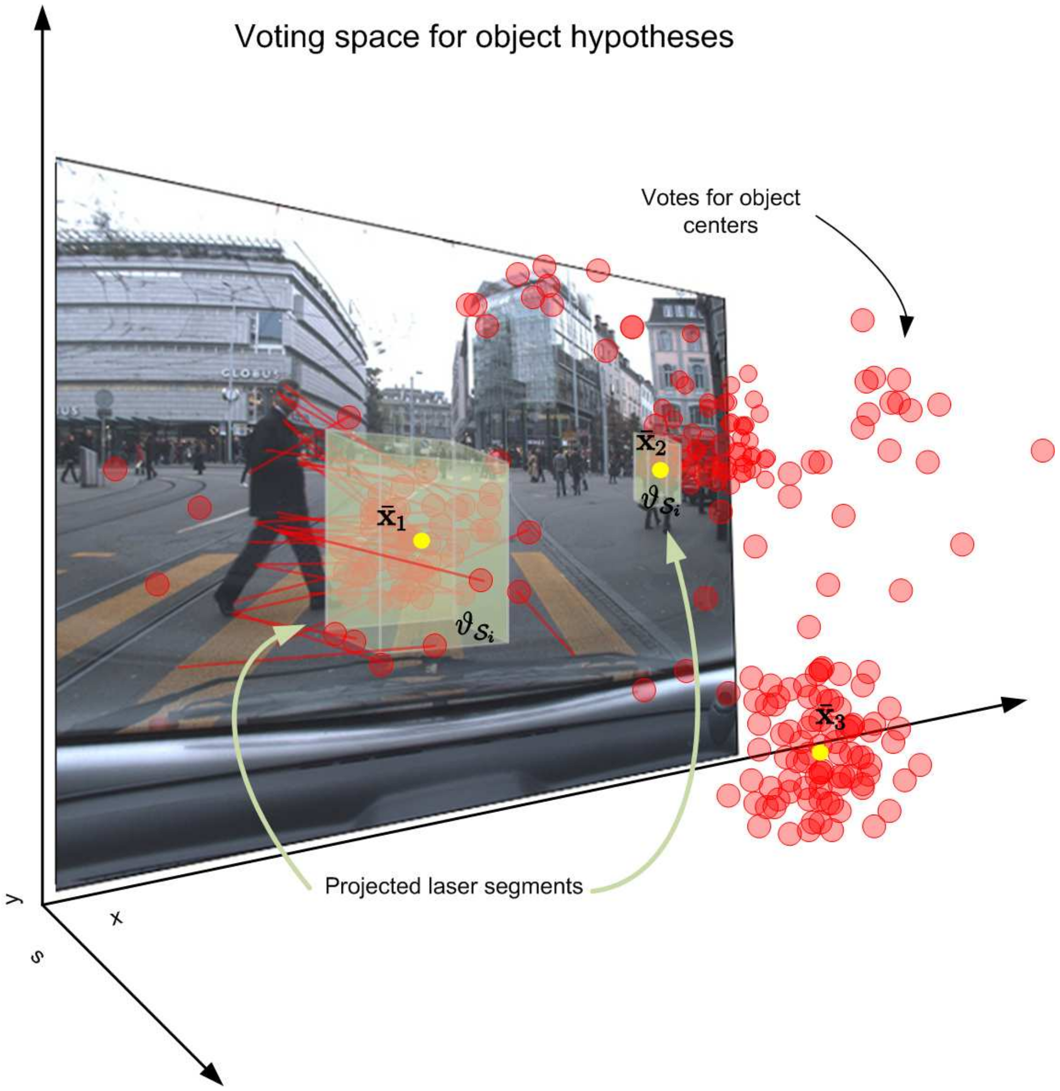

Figure 6: Early laser-camera fusion. Laser segments are projected into the visual-based object detection voting space as 3D boxes. Image detection hypotheses located into one of these regions are considered valid, the others are discarded. Votes are shown as red circles. Object hypotheses $\bar{\mathbf{x}}_i$ are shown in yellow. For clarity, features voting for the object centers (defined by position and scale) are shown only on the left pedestrian.

where $\vartheta^*_{\mathcal{S}}$ is a constant which is fixed beforehand. An image detection hypothesis is considered valid if it is found inside one of the 3D boxes that define the constraints of the voting space. A visual explanation can be seen in Fig. 6.

The last part of the early fusion step solves the data association problem between segments and corresponding image hypotheses. We assume that each segment belongs to a single object. For each segment we compute the distance and compute the hallucinated scale accroding to Eqn. (20). We solve the assignment problem in a greedy manner: given a segment $\mathcal{S}_i$, we assign from all valid image detection hypotheses found in the projected segment volume that one to the segment $\mathcal{S}_i$, which minimizes the absolute difference between the scale of the hypothesis and the hallucinated scale. The remaining processing of hypothesis selection for detection in camera images follows the technique explained in Section (4.4).

6.2 Combined detection using Kalman Filtering

The aim of multimodal object detection is to provide useful information to a navigation or a driver assistance module. For this reason, a natural output choice for our detector is to label laser segments with their class probability. The proposed fusion method combines the detectors' information and provides output that consists in laser segment positions and object category labels.

We use tracking as a mean of integrating class probabilities over time and as an additional algorithm output, to provide prediction information. We aim to design a reliable tracking method that does not rely on single data association hypotheses and that scales gracefully with the number of objects. Several methods have been developed in the tracking literature for handling complex data association at a high computational cost, including Multi-hypothesis tracking [Reid, 1979, Cox and Hingorani, 2002] and JPDA filters [Bar-Shalom and Li, 1995]. Our tracking algorithm is designed to be computationally inexpensive and copes well with the motion model of several kinds of object categories.

In contrast to cars, which have a comparably simple motion model given by the Ackermann model [Ackermann, 1818], pedestrians are much harder to describe with a single motion model: they can stop, suddenly turn on spot, invert their trajectory etc. Therefore, we use a pedestrian tracker in which each track is described by multiple Kalman Filters, each providing a different motion model. The advantage of this method is that the number of estimating filters scales linearly with the number of objects to track. Moreover, multiple hypotheses regarding object motions are produced for each time step. For this work we employed two kind of motion models, described by linear velocity and Brownian motion. The motivation for selecting Brownian motion is the ability to model sudden direction and speed changes, a condition that occurs especially in case of people tracking. Nevertheless, a constant velocity model, in short intervals, well approximates a variety of smooth curved trajectories. The proposed tracking technique is a way of combining tracking filters and it is very generic: other motion models, linear and non-linear, could be also used.

Tracks are managed by a *tracking manager* that solves data association, and creates or deletes tracks. We assume that each track is associated at most to one single segment. N is the number of laser segments present in a laser scan, R the number of tracks and M is the number of Kalman Filters present for each track, each with a different motion model. Data association, i.e. the problem of assigning laser segments to tracks, is solved in two steps. The first step is to compute which motion model to use for each track. In each track, the distance between the Kalman Filters (KF) prediction and the N laser segments centroids are computed. This process generates for each track M Mahalanobis distances for each observation [Mahalanobis, 1936]. In each track, the closest distance for each observation is taken and the KF generating that prediction is tagged. At the end of the first step of the association of laser segments, every track obtains a set of N distances from N observations.

The second step of data association is used to select which observation is assigned to which track. We want to assign N hypotheses to R tracks (where $N \neq R$). A rectangular matrix of size $R \times N$ is generated in which rows represent track indices and columns observation indices. The previously computed distances are inserted as values of the assignment matrix. The solution of the combinatorial minimal weight assignment has been found with the extension of Munkres' method for rectangular assignment matrices proposed by Bourgeois and Lassalle [1971]. If there are more segments than tracks, then $R - N$ new tracks are created. Instead, if more tracks than segments are present in a certain moment, the tracks that are not updated with a new observation are maintained until their variance in (x,y) reaches a fixed maximum threshold $\delta_{x,y}$.

We now give a mathematical formulation for the tracks and for the fusion of the detection outputs. We track cluster centroids in 2D range data using two KF, each with a different motion model:

$$\begin{aligned} \mathbf{x}_{m1} &= \Big((\hat{x}^{\mathcal{S}},\hat{y}^{\mathcal{S}}),(\dot{\hat{x}}^{\mathcal{S}},\dot{\hat{y}}^{\mathcal{S}}),(p_1,\ldots,p_C)\Big) && (22)\\ \mathbf{x}_{m2} &= \Big((\hat{x}^{\mathcal{S}},\hat{y}^{\mathcal{S}}),(p_1,\ldots,p_C)\Big), && (23) \end{aligned}$$

where $(\hat{x}^{\mathcal{S}},\hat{y}^{\mathcal{S}})$ are the coordinates of the cluster centroid, $(\dot{\hat{x}}^{\mathcal{S}},\dot{\hat{y}}^{\mathcal{S}})$ is its velocity and $p_1,\ldots,p_n$ are the probabilities of all C classes. The observation vector $\mathbf{z}(k)_i$, at time k, consists of the position of the cluster centroid and the category's probability estimates for each detection modality:

$$\mathbf{z}_i = \Big(\hat{x}_i^{\mathcal{S}},\hat{y}_i^{\mathcal{S}},(c_1,\ldots,c_n)^1,\ldots,(p_1,\ldots,p_C)^{\varsigma}\Big). \tag{24}$$

Here, $(\hat{x}^{\mathcal{S}},\hat{y}^{\mathcal{S}})$ is an observation of a cluster centroid and ς denotes the number of sensors. Each block $(p_1,\ldots,p_C)$ is the estimate given by the range or image based classifier.

The Kalman Filter is formulated by a prediction and an update step. Prediction at time k is computed by

$$\mathbf{x}(k)^{-}_{mi} = \mathbf{A}_{mi}\mathbf{x}(k-1)^{-}_{mi}. \tag{25}$$

We write the state matrices $\mathbf{A}_{mi}$ in the case of two motion models and two classes as

$$\mathbf{A}_{m1} = \begin{pmatrix} 1 & 0 & \Delta k & 0 & 0 & 0 \\ 0 & 1 & 0 & \Delta k & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad \mathbf{A}_{m2} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \tag{26}$$

If the matrix $\mathbf{V}^1$ indicates the state covariance and $\mathbf{V}^2$ the sensor covariance, we compute

$$\mathbf{P}(k)^-_{mi} = \mathbf{A}_{mi}\mathbf{P}(k-1)\,\mathbf{A}^T_{mi} + \mathbf{V}^1. \tag{27}$$

The tracker manager selects which KF of each track is closer to the observation $\mathbf{z}_i$. Then it solves the data association between the winning KF of each track and observations using the assignment optimization proposed by Bourgeois and Lassalle [1971]. The observations are assigned to the tracks and the filters are updated. The observation is used to update all the filters of the track. The update step is calculated by computing the Kalman gain $\mathbf{G}$ and updating $\mathbf{x}(k)_{mi}$ and the covariance matrix $\mathbf{P}$, i.e.

$$\begin{aligned} \mathbf{K}_{mi} &= \mathbf{P}(k)^-_{mi}\mathbf{G}^T\left(\mathbf{G}\,\mathbf{P}(k)^-_{mi}\,\mathbf{G}^T + \mathbf{V}^2\right)^{-1} && (28) \\ \mathbf{P}(k)_{mi} &= (I - \mathbf{K}_{mi}\mathbf{G})\,\mathbf{P}(k)^-_{mi} && (29) \\ \mathbf{x}(k)_{mi} &= \mathbf{x}(k)^-_{mi} + \mathbf{K}_{mi}\left(\mathbf{z}(k)_a - \mathbf{G}\mathbf{x}(k)^-_{mi}\right), && (30) \end{aligned}$$

where $\mathbf{z}(k)_a$ represents the assigned observation vector to the track. The matrix $\mathbf{G}$ models the mapping from states to the predicted observation and is defined as $\mathbf{G} = (\mathbf{G}^T_x\mathbf{G}^T_{s1}\ldots\mathbf{G}^T_{sC})^T$, where $\mathbf{G}_x$ maps to pose observations and the $\mathbf{G}_{s1}$ map to class probabilities per sensor. For example, for one laser, one camera and constant velocity we have:

$$\mathbf{G}_{s1} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix} \qquad \mathbf{G}_{s1} = \mathbf{G}_{s2} = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}. \tag{31}$$

7 Experimental Results

In this section, we present experimental results and quantitative comparisons with other techniques in order to validate our method.

7.1 Experimental Platform

To acquire the data, we employed our urban mobile platform *Smartter*. The robot is based on a standard Smart car that has been equipped with distance laser sensors, cameras, a differential

Num Frames	1675
Laser range data resolution	0.25deg
Image resolution	640×480px
Laser positioning	horizontal, 48cm from ground
Camera lens	Telelens, 45deg f.o.v.
Num of car samples	510
Num of people samples	376

Table 1: Urban Scenario testing dataset, collected in downtown Zürich, Switzerland

GPS unit, an Inertial Measurement Unit (IMU), an optical gyroscope and several processing computers. For this work, we acquired data with a camera equipped with a telelens and a 2D laser range finder mounted in front. The camera was mounted on a metal rig on the rooftop of the vehicle and the logging system has been optimized to reduce frame drops.

7.2 Real World Dataset: Urban Scenario

We evaluated our technique on a challenging urban scenario dataset. We set the laser angular resolution to 0.25 degrees in order to obtain a high resolution laser dataset. Data is collected inside Zurich, Switzerland in a loop of circa 1*km* length to retrieve cars and pedestrians in a real busy urban environment. We synchronized camera and laser data for a total of 1675 frames. The imagery is manually labeled with rectangle boxes indicating pedestrians and cars. Annotations in images are marked if at least half of an object is shown or the object width in the image is greater than 80 pixels. Laser range data has been manually labeled by using associated image frames as reference for the ground truth. Labeling is obtained by manually selecting and assigning a class label to the segments in the range data. A suite of MATLAB scripts has been used to simplify this process.

7.3 ISMe image detector training

Several ISM codebooks need to be trained due to the complexity of the multiclass (cars, pedestrians) classification task. Experience shows [Leibe et al., 2007] that lateral views of pedestrians generalize well to front/back views. Therefore, we used a set composed of 400 images of persons with a height of 200 pixels at different positions, dressed with different clothing and accessories such as backpacks and hand bags in a typical urban environment. The category ’car’ has been learned from 7 different viewpoints as in Leibe et al. [2007] (see also Figure 7, left). 200 training images are used for each view. Car codebooks are learned using Shape Context (SC) descriptors [Belongie et al., 2002] at Hessian-Laplace interest points [Mikolajczyk and Schmid, 2005]. The pedestrian codebook uses lateral views and SC descriptors at Hessian-

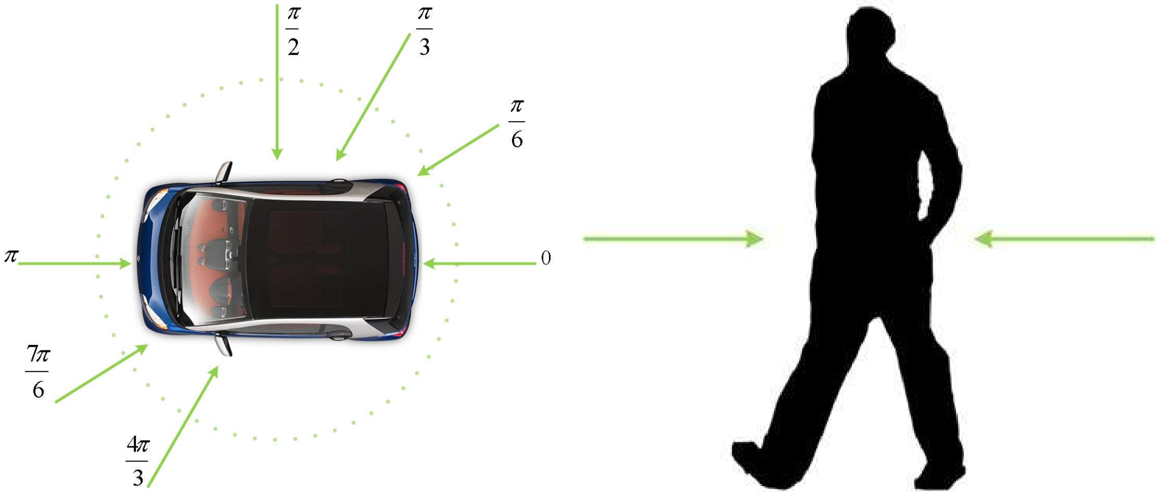

Figure 7: **Left:** For car classification, we use codebooks from 7 different views. For training, mirrored images are included for each view to obtain a wider coverage. **Right:** For pedestrians we use a codebooks of side views with mirroring. Lateral views have sufficient information to generalize frontal/back views.

Laplace and Harris-Laplace interest points for more robustness. We selected SC due to their low dimensionality (36D): this shortens the time for feature extraction, for the agglomerative clustering of the codebook generation and for feature matching with codebooks. In the work of Leibe et al. [2006], the authors compare several descriptors for object detection and they show that SC descriptors are very good features for object detection.

7.4 Boosted CRF range detector training

Our laser training data consists of 600 annotated scans containing pedestrians, cars and background randomly sampled from a typical urban scenario. 5158 car data points, 2379 people data points and 25251 background labeled points have been used for training. There is no distinction of car views in the laser data as the variation in shape is low. The range data is limited to a maximum range of 15*m*. As a first step, the AdaBoost classifier of range data features is trained on this set. Then we use the output of the trained classifier to produce node feature values for the CRF. Then, the CRF is trained in order to set the node and edge features weights.

7.5 Quantitative and Qualitative evaluation

In this section we present results in form of precision-recall curves. They summarize the complete performance of a classifier:

$$\text{Precision} = \frac{\text{TruePos}}{\text{TruePos} + \text{FalsePos}} \quad \text{Recall} = \frac{\text{TruePos}}{\text{TruePos}+\text{FalseNeg}} \tag{32}$$

Precision-recall curve has the advantage of computing a classifier performance measure without knowing the number of true negatives. Specially in case of image and range data classification, setting this number can be ambiguous because the quantity of possible true negatives in such data is not easy to define.

We run a comparison of the proposed multiclass image detection algorithms with our previous work [Spinello et al., 2008b], as shown in Figure 13. Our vision based multiclass detection, named ISMe2.0 in the plots, is compared to the standard ISM, our previous single class detector ISMe1.0 and with an AdaBoost detector trained on Haar features (ABH). We can see that our method yields the best results. It is important to see that the multiclass method obtains higher recall values than the previous ISMe1.0, mostly due to the refinements introduced in the hypothesis selection step, namely the object subparts and the shape templates.

We then run the system for the challenging Urban Scenario dataset. Pedestrian detection with camera is shown in Fig. 8-left.

In the evaluation of results we compare the performance of several detectors by using equal error rate (EER) error metric on a precision-recall graph. EER is a measure to compare the accuracy of classifier. This measure is often used, especially in biometrics [K.P. Li, 1988] and in computer vision [Leibe et al., 2005]. In general, the classifier with the lowest EER is most accurate. EER is the point in which false positive rate and false negative rate have the same value. The lower the EER, the more accurate the system is considered to be. The higher the diagonal crossing point in the precision-recall curve, the lower EER, the less the errors computed by the classifier.

We compared our image detector with respect to a Haar-AdaBoost based classifier and, in case of the pedestrian detector, with the Histogram of Oriented Gradients technique developed by Dalal and Triggs [2005]. In case of HOG and ABH we used the early fusion technique explained in Section 6.1 in order to reduce the image search space. Our multiclass detector, shortly named ISMe, clearly outperforms the other methods. Precision at equal error rate (EER) is: 60.01% for ISMe, 52.21% for HOG, 11.17% for ABH. In general, if one is willing to accept a high rates of false positives, the ISMe detector could achieve a > 70% Recall. At that values the difference with respect to the other methods is even more evident. We then evaluated the laser based detector for pedestrian detection in Figure 8-right. There we show a comparison between the Boosted CRF and a standard AdaBoost classification of JDC segments (AJDC) in order to visualize the introduced performance enhancements. AJDC classifies JDC segments regardless of the neighboorhood state. It is interesting to notice that the consideration of the segments' neighborhood in the CRF plays an important role in the ability to increase the

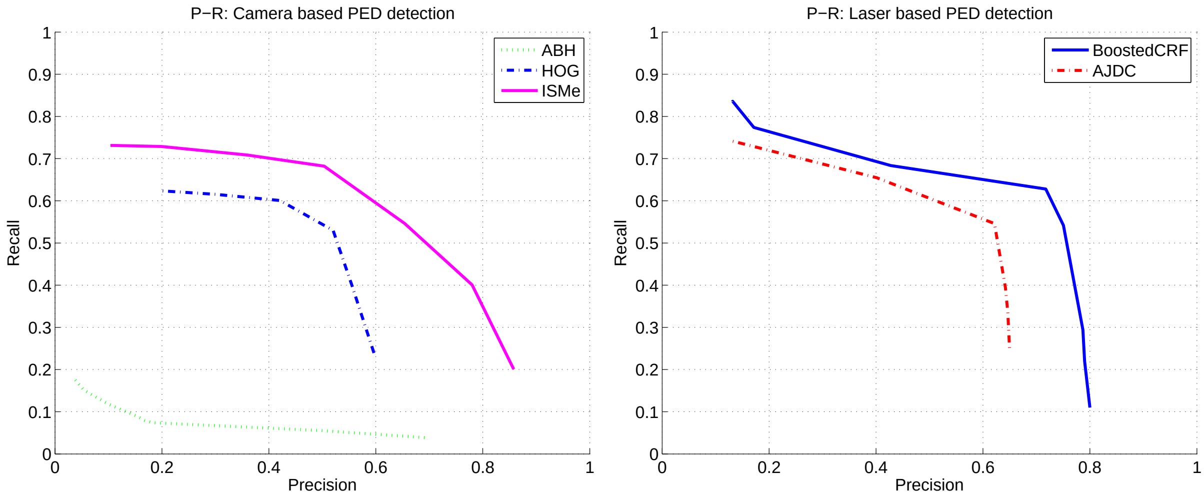

Figure 8: Quantitative evaluation for pedestrian detection. Our approach outperforms the other methods for both sensor modalities. The image based detection is compared with Histogram of Oriented Gradients detector (HOG) and an AdaBoost classifier using Haar features (ABH). We show a comparison between Boosted CRF and AdaBoost classification of JDC segments (AJDC) in order to visualize the introduced performance enhancements.

detection rate and reduces the number of false positives, the AJDC curve is always below the CRF one and it decreases earlier than the CRF curve. In this case precision at EER is 64.23% for the CRF and 57.09% for AJDC.

We then evaluated the performance of our system in case of car detection (see Figure 10). The ISMe car image detector outperforms the ABH detector. The latter has been trained on trunks, sides and frontal views of cars. It is important to remark that the results shown in Figure 10-left are averaged between the 7 car views of ISMe. the Equal Error Rate is crossed at 72.54% for ISMe and 18.93% for ABH. The performance of the laser based classifier is compared with AJDC in Fig. 10-right and also in this case CRF has better results with respect to AJDC. Precision at EER is 74.89% for CRF and 70.57% for AJDC. It is interesting to notice that cars are in general easier to detect with respect to pedestrians. Intuitively, cars are rigid objects with much less geometrical and visual variability than visually complex pedestrians.

Tracking and fusion for the pedestrian category is evaluated in Fig. 9. We show the precision-recall graph and a ’Recall-false positives per frame’ plot in order to show the performance increase. It is interesting to see in the plot of Fig. 9-left that the camera and laser detectors are very complementary sources of information: their combined contribution allows to have a fused detection that is higher that each single one; this phenomenon is even more evident when precision is low. The tracked and fused precision at EER is 69.8%. In Figure 9-right we show that we improved also that: by fixing a certain false positive rate per frame, we obtain

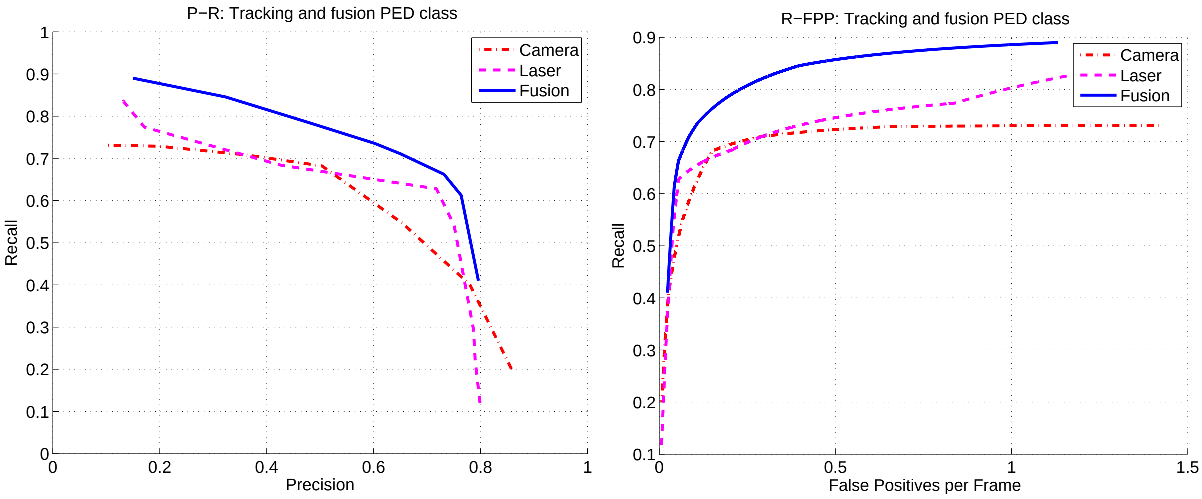

Figure 9: Quantitative evaluation of tracking and fusion for pedestrian detection. Precision-Recall graph (left) and 'Recall-false positives per frame' show that the fusion method enhances the results of single classifiers.

a higher Recall value. Tracking and fusion for cars is shown in Fig. 11. Similar conclusions to the sensor fusion on pedestrians could be given. Tracking allows a better detection rate than each single classifier and a reduced number of false positives per frame; precision at EER is 78.4%. This value of precision is significantly higher than the pedestrian category due to the higher performances of vision and laser detectors.

From this experiments we can draw some interesting conclusions. Image and range data are two very different sensor modalities, with very different characteristics. With this experiments we proved that image and range based detectors can be combined for obtaining a fused detector that is more robust than its components. Range data has the advantage of a precise and instantaneous target localization and it helps to distinguish objects that have a low image information content, for instance people in shadow areas, or partial views of cars. Image, instead, plays an important role when range data is ambiguous, for instance when a person is observed from the side or in presence of clutter. Both of this examples show how single sensor modalities could fail and how the multimodal fusion overcomes this flaws. Moreover it is interesting to notice, that in case of limited visibility, poor/no light conditions or camera failure, this approach still produces a usable output, see for instance Figure 12-middle or Figure 12-bottom.

Certainly, this approach presents shortcomings. The technique is limited to the range of 15m due to sparsity of the retrieved laser data points. At such distance cars and specially people are described by too few points to obtain good range data classification results. Severe street slopes could also contribute to *short-sightedness* of the fused detector. This aspect has been addressed in a previous work Spinello et al. [2008b] by using 3D ground plane estimate

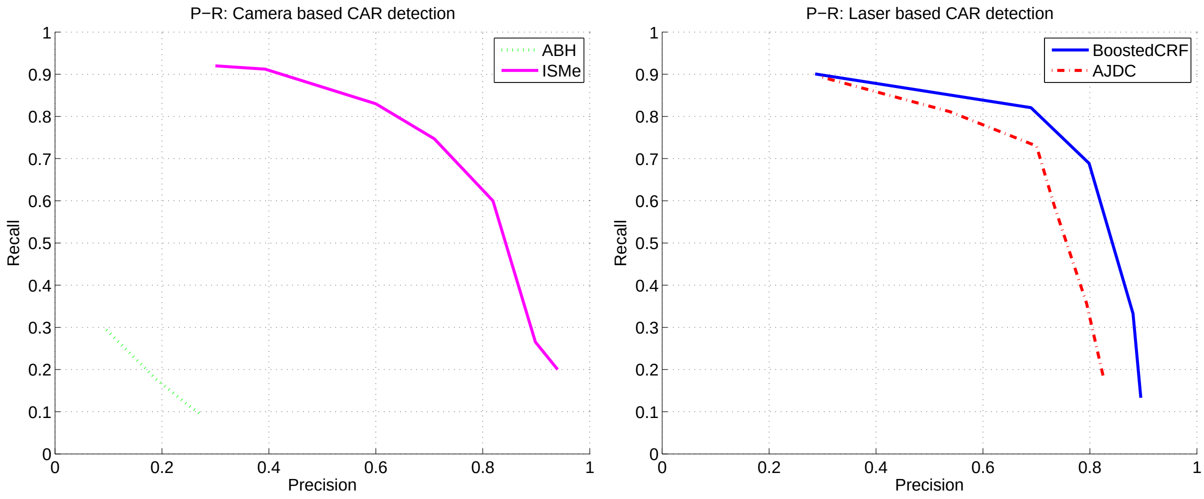

Figure 10: Quantitative evaluation for car detection. Our approach outperforms the other methods for both sensor modalities. The image based detection is compared with an AdaBoost classifier using Haar features (ABH). We show a comparison between Boosted CRF and AdaBoost classification of JDC segments (AJDC) in order to visualize the introduced performance enhancements.

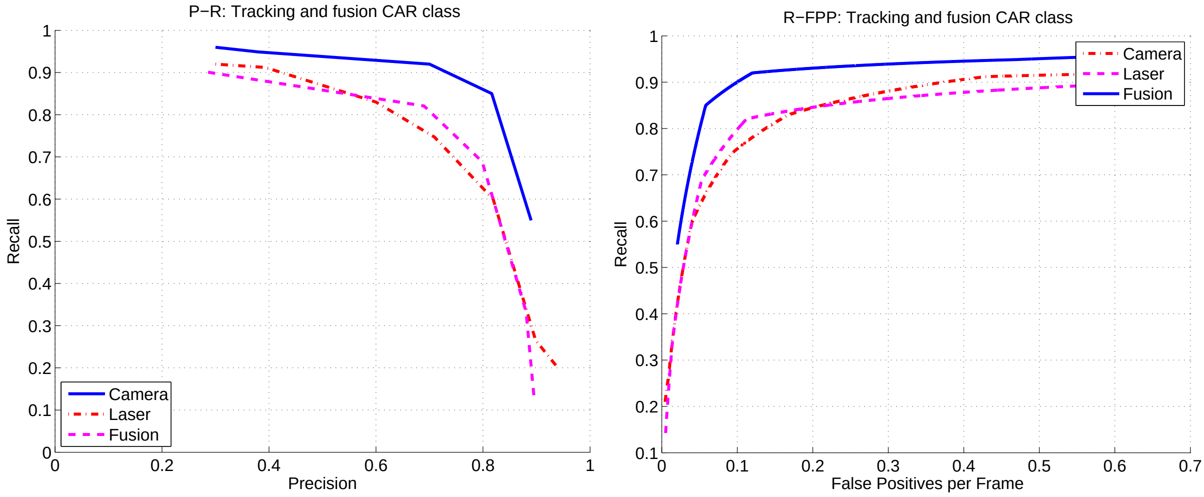

Figure 11: Quantitative evaluation of fusion for car detection. Precision-Recall graph (left) and ’Recall-false positives per frame’ show that the fusion method enhances the results of single classifiers.

with a 3D laser.

Some qualitative results are shown in Figure 12 where a passing car and a crossing pedestrian are correctly detected and tracked. It is important to notice that even though images and laser data show very low contrast, partial occlusions and clutter, the system manages to detect and track the objects in the scene. For a video extracted from the experiments see Extension 1 (Appendix A).

8 Conclusions

We have presented a method to reliably detect and track multiple object classes in outdoor scenarios using vision and 2D laser range data. We showed that the overall performance of the system is improved using a multiple-sensor system. We have introduced several extensions to the ISM based image detection to cope with multiple classes. We showed that laser detection based on Boosted CRFs performs better than a simpler AdaBoost classifier and presented tracking results on combined data. Finally, we showed the usefulness of our approach through extended experimental results and comparisons on real-world data.

Future developments of this research are concerned specially with the integration of long range people detection. People at long range are described by only a few pixels in the image and few to none laser points. The idea is to integrate small scale detection methods [Spinello et al., 2009] in the multimodal system by considering a more advanced tracking able to cope with very unreliable hypotheses. Other research directions involve the development of robust data association filters, like MHT or JPDAF, adapted to the multimodal detection problem.

9 Acknowledgement

This work has been partly supported by European Union under contract numbers BACS-FP6-IST-027140 and EUROPA-FP7-231888.

A Index to Multimedia Extensions

The multimedia extensions to this article are at: *http://www.ijrr.org*.

Extension	Type	Description
1	Video	Multimodal detection and tracking of people and cars

References

R. Ackermann. Improvements on axletrees applicable to four-wheeled carriages. In *Patent Nr. 4212*, 1818.

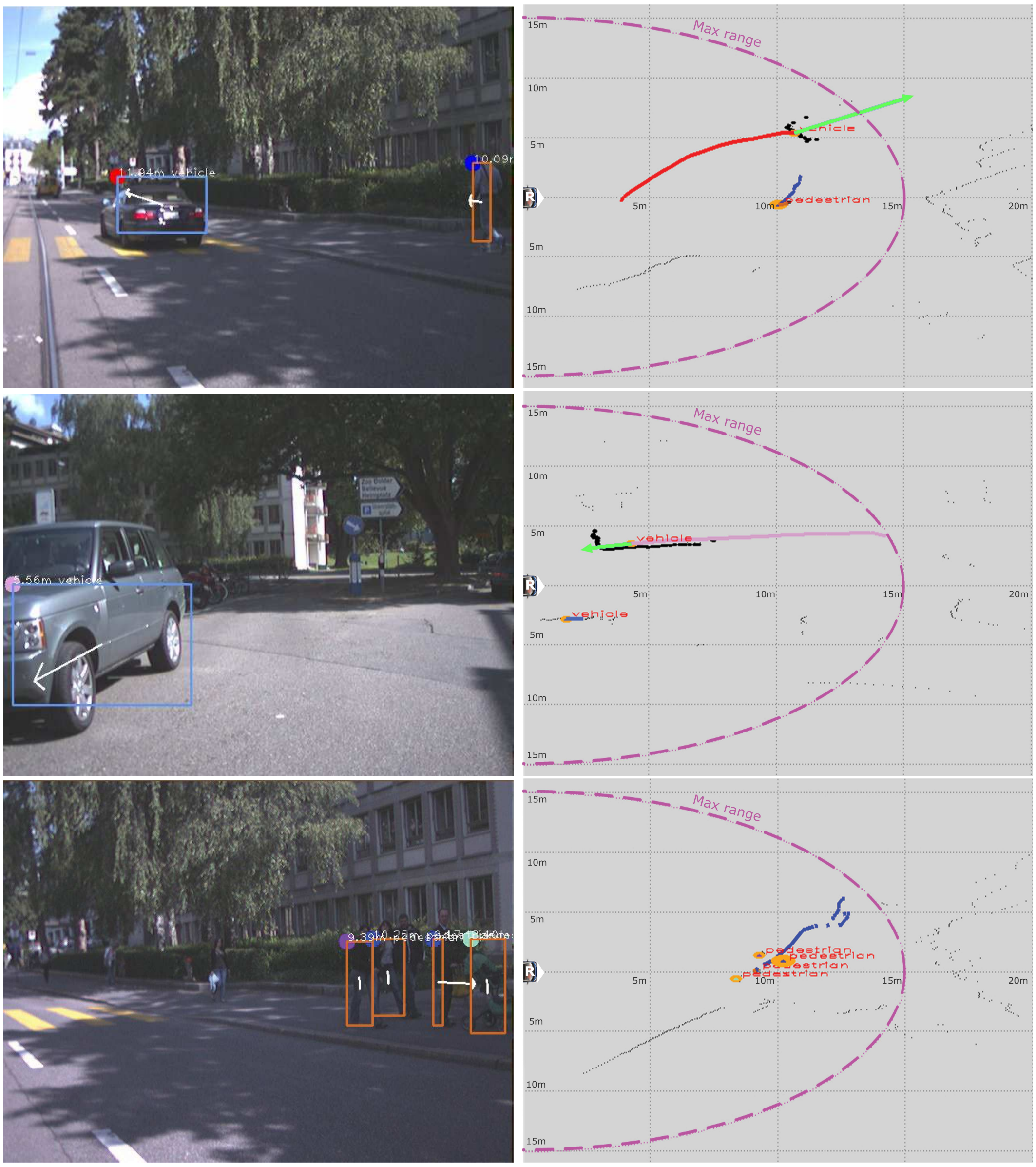

Figure 12: Cars and pedestrian detected and tracked under occlusion, clutter and partial views. In the camera images, left column, blue boxes indicate car detections, orange boxes pedestrian detections. The colored circlise on the upper left corner of each box is the track identifier. Tracks are shown in color in the right column and plotted with respect to the robot reference frame. Green vectors show direction of motion for cars.

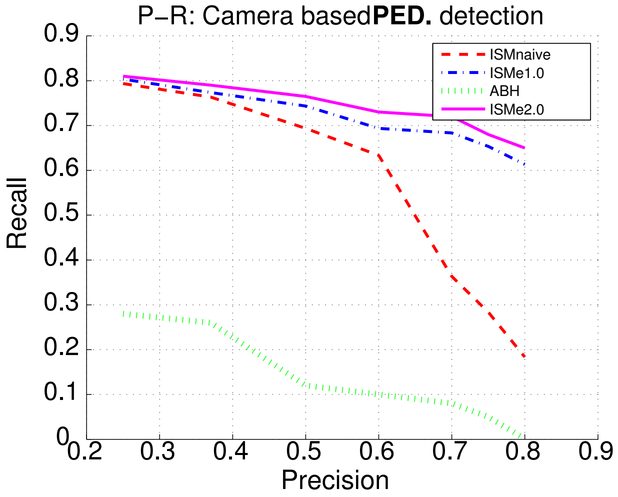

Figure 13: Quantitative evaluation for pedestrian detection. From left to right we compared the newly introduced multiclass technique with other approaches. The multiclass image based detector, ISMe2.0 is evaluated for the pedestrian category and it is compared with standard ISM (ISMnaive), a previous single class pedestrian detector ISMe (ISMe1.0) and AdaBoost with Haar features (ABH) classifier.

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

K. O. Arras, S. Grzonka, M. Luber, and W. Burgard. Efficient people tracking in laser range data using a multi-hypothesis leg-tracker with adaptive occlusion probabilities. In *IEEE Int. Conf. on Rob. and Autom. (ICRA)*, 2008.

Y. Bar-Shalom and X. Li. *Multitarget-Multisensor Tracking: Principles and Techniques*. YBS Publishing, 1995.

S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Trans. on Pattern Analysis & Machine Intelligence*, 24(4), 2002.

G. Borgefors. Hierarchical chamfer matching: A parametric edge matching algorithm. *IEEE Trans. on Pattern Analysis & Machine Intelligence*, 10(6), 1988.

F. Bourgeois and J. C. Lassalle. An extension of the munkres algorithm for the assignment problem to rectangular matrices. *Communications of the ACM*, 14(12), 1971.

D. Comaniciu, V. Ramesh, and P. Meer. The variable bandwidth mean shift and data-driven scale selection. In *IEEE Int. Conf. on Computer Vision (ICCV)*, 2001.

I. Cox and S. L. Hingorani. An efficient implementation of reid's multiple hypothesis tracking algorithm and its evaluation for the purpose of visual tracking. *IEEE Trans. on Pattern Analysis & Machine Intelligence*, 18(2), 2002.

J. Cui, H. Zha, H. Zhao, and Shibasaki. Tracking multiple people using laser and vision. In *IEEE Int. Conf. on Intell. Rob. and Systems (IROS)*, 2005.

N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *IEEE Conf. on Comp. Vis. and Pat. Recog. (CVPR)*, 2005. ISBN 0-7695-2372-2.

B. Douillard, D. Fox, and F. Ramos. Laser and vision based outdoor object mapping. In *Robotics: Science and Systems (RSS)*, 2008.

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

P. Felzenszwalb and D. Huttenlocher. Efficient matching of pictorial structures. In *IEEE Conf. on Comp. Vis. and Pat. Recog. (CVPR)*, 2000.

A. Fod, A. Howard, and M. J. Matarić. A laser-based people tracker. In *IEEE Int. Conf. on Rob. and Autom. (ICRA)*, 2002.

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

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

D. Hähnel, R. Triebel, W. Burgard, and S. Thrun. Map building with mobile robots in dynamic environments. In *IEEE Int. Conf. on Rob. and Autom. (ICRA)*, 2003.

S. Ioffe and D. A. Forsyth. Probabilistic methods for finding people. *Int. Journ. of Computer Vision*, 43(1), 2001.

J. P. K.P. Li. Normalizations and selection of speech segments for speaker recognition scoring. In *Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, 1988.

J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmentation and labeling sequence data. In *Int. Conf. on Machine Learning (ICML)*, 2001.

B. Lau, K. O. Arras, and W. Burgard. Tracking groups of people with a multi-model hypothesis tracker. In *IEEE Int. Conf. on Rob. and Autom. (ICRA)*, 2009.

B. Leibe, E. Seemann, and B. Schiele. Pedestrian detection in crowded scenes. In *IEEE Conf. on Comp. Vis. and Pat. Recog. (CVPR)*, 2005.

B. Leibe, K. Mikolajczyk, and B. Schiele. Segmentation based multi-cue integration for object detection. In *British Machine Vision Conf. (BMVC)*, 2006.

B. Leibe, N. Cornelis, K. Cornelis, and L. V. Gool. Dynamic 3d scene analysis from a moving vehicle. In *IEEE Conf. on Comp. Vis. and Pat. Recog. (CVPR)*, 2007.

D. Liu and J. Nocedal. On the limited memory BFGS method for large scale optimization. *Math. Programming*, 45(3, (Ser. B)), 1989.

S. P. Lloyd. Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28 (2), 1982.

M. Luber, K. O. Arras, C. Plagemann, and W. Burgard. Classifying dynamic objects: An unsupervised learning approach. In *Robotics: Science and Systems (RSS)*, 2008.

P. Mahalanobis. On the generalised distance in statistics. In *Proceedings National Institute of Science, India*, volume 2, 1936.

K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Trans. on Pattern Analysis & Machine Intelligence*, 27(10), 2005.

L. Navarro-Serment, C. Mertz, and M. Hebert. Pedestrian detection and tracking using three-dimensional ladar data. In *Intern. Conf. of Field and Service Robotics (FSR)*, July 2009.

C. Papageorgiou and T. Poggio. A trainable system for object detection. *Int. Journ. of Computer Vision*, 38(1), 2000.

A. Petrovskaya and S. Thrun. Model based vehicle tracking for autonomous driving in urban environments. In *Robotics: Science and Systems (RSS)*, 2008.

R. Pless and Q. Zhang. Extrinsic calibration of a camera and laser range finder. In *IEEE Int. Conf. on Intell. Rob. and Systems (IROS)*, 2004.

R. B. Potts. Some generalized order-disorder transformations. *Cambridge Phil. Society*, 48, 1952.

C. Premebida, O. Ludwig, and U. Nunes. Lidar and vision-based pedestrian detection system. *Journal of Field Robotics*, 26(9), 2009.

F. Ramos, D. Fox, and H. Durrant-Whyte. CRF-matching: Conditional random fields for feature-based scan matching. In *Robotics: Science and Systems (RSS)*, 2007.

D. Reid. An algorithm for tracking multiple targets. *IEEE Transactions on Automatic Control*, 24(6), 1979.

M. Scheutz, J. Mcraven, and G. Cserey. Fast, reliable, adaptive, bimodal people tracking for indoor environments. In *IEEE Int. Conf. on Intell. Rob. and Systems (IROS)*, 2004.

D. Schulz. A probabilistic exemplar approach to combine laser and vision for person tracking. In *Robotics: Science and Systems (RSS)*, 2006.

D. Schulz, W. Burgard, D. Fox, and A. Cremers. People tracking with mobile robots using sample-based joint probabilistic data ass. filters. *Int. Journ. of Robotics Research (IJRR)*, 22(2), 2003.

G. Schwarz. Estimating the dimension of a model. *Annals of Statistics*, 6(2), 1978.

L. Sirovich and M. Kirby. Low-dimensional procedure for the characterization of human faces. *Journ. Optical Society of America. A*, 4(3), 1987.

L. Spinello and R. Siegwart. Human detection using multimodal and multidimensional features. In *IEEE Int. Conf. on Rob. and Autom. (ICRA)*, 2008.

L. Spinello, R. Triebel, and R. Siegwart. Multimodal people detection and tracking in crowded scenes. In *Proc. of the AAAI Conf. on Artificial Intelligence*, 2008a.

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

L. Spinello, A. Macho, R. Triebel, and R. Siegwart. Detecting pedestrians at very small scales. In *IEEE Int. Conf. on Intell. Rob. and Systems (IROS)*, 2009.

L. Spinello, K. O. Arras, R. Triebel, and R. Siegwart. A layered approach to people detection in 3d range data. In *Proc. of the AAAI Conf. on Artificial Intelligence*, 2010.

D. Streller, K. C. Fuerstenberg, and K. Dietmaye. Vehicle and object models for robust tracking in traffic scenes using laser range images. In *Int. Conf. on Int. Tranp. Systems (ITSC)*, 2002.

E. A. Topp and H. I. Christensen. Tracking for following and passing persons. In *IEEE Int. Conf. on Intell. Rob. and Systems (IROS)*, 2005.

P. Viola, M. J. Jones, and D. Snow. Detecting pedestrians using patterns of motion and appearance. In *IEEE Int. Conf. on Computer Vision (ICCV)*, 2003.

S. Wender and K. Dietmayer. 3D vehicle detection using a laser scanner and a video camera. *Intelligent Transport. Systems (IET)*, 2008.

J. Xavier, M. Pacheco, D. Castro, A. Ruano, and U. Nunes. Fast line, arc/circle and leg detection from laser scan data in a player driver. In *IEEE Int. Conf. on Rob. and Autom. (ICRA)*, 2005.

Z. Zhang. Flexible camera calibration by viewing a plane from unknown orientations. In *IEEE Int. Conf. on Computer Vision (ICCV)*, 1999.

L. Zhao and C. Thorpe. Qualitative and quantitative car tracking from a range image sequence. In *IEEE Conf. on Comp. Vis. and Pat. Recog. (CVPR)*, 1998.

W. Zheng and L. Liang. Fast car detection using image strip features. In *IEEE Conf. on Comp. Vis. and Pat. Recog. (CVPR)*, 2009.

Q. Zhu, M. C. Yeh, K. T. Cheng, and S. Avidan. Fast human detection using a cascade of histograms of oriented gradients. In *IEEE Conf. on Comp. Vis. and Pat. Recog. (CVPR)*, 2006.

Z. Zivkovic and B. Kröse. Part based people detection using 2D range data and images. In *IEEE Int. Conf. on Intell. Rob. and Systems (IROS)*, 2007.