Unsupervised Classification of Dynamic Obstacles in Urban Environments

Roman Katz
Australian Centre for Field Robotics
The University of Sydney
Sydney, NSW 2006, Australia
r.katz@acfr.usyd.edu.au

Juan Nieto
Australian Centre for Field Robotics
The University of Sydney
Sydney, NSW 2006, Australia
j.nieto@acfr.usyd.edu.au

Eduardo Nebot
Australian Centre for Field Robotics
The University of Sydney
Sydney, NSW 2006, Australia
nebot@acfr.usyd.edu.au

Abstract

This paper presents a solution to the problem of unsupervised classification of dynamic obstacles in urban environments. A *track-based* model is introduced for the integration of 2D laser and vision information that provides a robust spatio-temporal synthesis of the sensed moving obstacles and forms the basis for suitable algorithms to perform unsupervised classification by clustering. This work presents various contributions in order to achieve accurate and efficient performance, initially using laser tracks for classification, and then incorporating visual tracks to the model. A procedure is proposed for accurate unsupervised classification of dynamic obstacles using a *laser stamp* representation of the tracks. Laser data is then integrated with visual information through a single-instance *visual stamp* representation, which is finally extended using a multiple instance framework to robustly deal with challenges associated with perception in real-world scenarios. The proposed algorithms are extensively validated with a simulated environment. Experiments with a research vehicle in an urban environment demonstrate the performance of the approach with real data. The experimental results reach an accuracy of over 92% for obstacle classification, finding the clusters that correspond to the main obstacle classes in the data.

1 Introduction

Accurate classification of obstacles from a moving vehicle is a vital component in any architecture developed to achieve some level of autonomy or to provide situation awareness information to drivers. In these scenarios, the considered obstacle classes usually determine different responses or levels of assessment related to the situation. Class information can be integrated within the global navigation architecture, for example, in obstacle avoidance, mapping or tracking modules. In assistance systems for commercial cars, classes can be used to trigger the corresponding alarms or actions.

The 2007 Urban Darpa Grand Challenge competition (DARPA, 2007) is a clear example of the importance and difficulty of obstacle classification for fully autonomous architectures. Although extensive research has

been undertaken towards obtaining accurate classification through on-board sensors, the competition proved to be "challenging" and only 6 teams out of the initial 89 teams qualified and successfully completed the final course. The most demanding situations arose in scenarios populated with dynamic objects perceived from an observer that was also moving. A consistent understanding of the world that includes and characterizes the dynamic nature of the obstacles is relevant beyond this particular competition. This capability has strong implications for any robotic system that performs perception tasks in real-world unstructured environments.

Classification of obstacles is one of the main challenges towards increasing safety in non-autonomous road vehicles. In this case, the aim of perception is to provide complementary information to a driver in control of the vehicle. Great effort has been concentrated on the development of safety technologies for *intelligent transportation systems* (ITS), including systems such as *adaptive cruise control* (ACC) (Yamamura et al., 2001), *advanced drive assistance systems* (ADAS) (Petersson et al., 2006) and *collision mitigation* (CM) architectures (Jansson, 2005). As new and cheaper communication technologies become available, there has been an increase in vehicle-to-vehicle and vehicle-to-infrastructure communication systems (Worral, 2009), where data is shared to plan and react accordingly and avoid risky situations. These systems are constrained by the coverage of radio communication and infrastructure, and vehicle-centered sensing approaches are still needed in rapidly changing and uncontrolled scenarios such as urban environments.

In spite of the various safety improvements achieved in the automotive industry in the last decade, there still exists a significant number of traffic incidents. Statistics extracted from the 2008 USA Traffic Safety Annual Assessment (NHTSA, 2008) show that the great majority of deaths and injuries involved cars, trucks, bikes, and pedestrians, for totals of over 37 thousand killed and 2.3 million injured. The number of incidents involving dynamic obstacles reaches an average of 70% over different types of vehicles, which highlights the importance of dealing with the classification of dynamic obstacles for safety navigation in urban vehicles.

This paper is concerned with the unsupervised classification of dynamic obstacles in urban environments using 2D laser and visual data. A *track-based* model is used as the basic representation within the proposed architecture that provides robust spatio-temporal synthesis of the sensed moving obstacles and forms the basis for algorithms to perform unsupervised classification by clustering. This model effectively considers the information from each of the obstacles' tracks, i.e., sequences of laser segments and images that are extracted from each track. The diagram in Figure 1 presents the proposed processing architecture with references from each component to the corresponding section in the paper.

The structure of the paper is as follows. Related work and contributions are considered in Section 2. The use of laser tracks for unsupervised classification of dynamic obstacles is presented in Section 3. The proposed approach is based on the formulation of *laser stamps* to capture shape information from the sensed dynamic objects and a laser stamp similarity measure suitable for similarity-based clustering. Clustering methods are explored and an extension of *affinity propagation* (AP) algorithm is derived to attain efficient clustering using laser stamps. Section 4 integrates visual track information for unsupervised classification. The scheme builds on the laser stamp representation and incorporates visual information for improving the clustering. A learning method based on *positive-only learning* (PL) that exploits *a priori* clustering given by the laser is introduced to compute visual similarity. Section 5 presents an approach based on *multiple instance learning* (MIL) to extend the visual model by utilizing full sequences of images. Instead of describing tracks by one representative image only, complete sequences are now considered, allowing the architecture to adapt to more challenging real-world scenarios. Experimental results are presented in Section 6 employing data collected in urban environments using a research vehicle equipped with a 2D Sick laser and a high-resolution color monocular camera. Discussion on lessons learned and conclusions are finally included in Section 7 and Section 8.

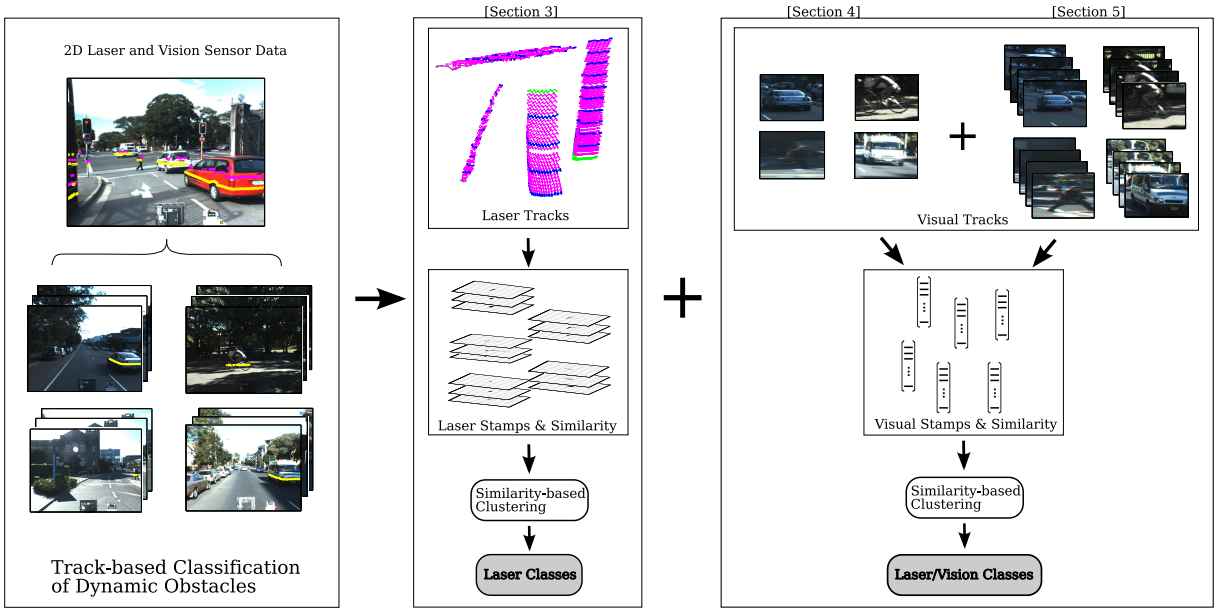


Figure 1: Processing pipeline of the proposed track-based architecture. Each module presents a different processing component that incrementally includes additional capabilities to the model. Module (a) illustrates the input sensor data that includes 2D laser and vision information. The modules in (b) (detailed in Section 3) use laser tracks for unsupervised classification. The components in (c) integrate visual track information to the architecture with increasing complexity. Section 4 uses single-instance visual stamps, whereas Section 5 expands the visual model by utilizing full sequences of images.

2 Related Work and Contributions

A large body of work has been presented on unsupervised classification using laser and vision. A vast majority of the work using vision only has been mostly aimed to perform automatic discovery of classes for visual recognition tasks. The work in (Burl et al., 1998), for instance, solves the model learning stage using *expectation-maximization* (EM) (Dempster et al., 1977). A constellation of features configuration (Fischler and Elschlager, 1973) is utilized and promising subsets of features and positions are evaluated in an iterative scheme, generating a model that minimizes a log-likelihood formulation. The work in (Weber et al., 2000) builds on this scheme proposing complementary unsupervised segmentation and feature extraction procedures for complete classification with no manual intervention. Highly textured regions are identified for stable feature extraction, and a clustering scheme that favors large clusters of features that correspond to the objects (rather than the background) is introduced to perform automatic segmentation and feature selection.

Specific techniques when using video sequences, rather than still images, have also been addressed by different researchers, in what is often called *learning from tracking* (Ramanan and Forsyth, 1999). The main idea behind these approaches is to capture dependencies between parts or features locally and over time. The approach in (Leordeanu and Collins, 2005) models the temporal correlation of parts belonging to the same objects with respect to the independent behavior of the rest of unrelated parts. The work from (Stauffer and Grimson, 2000) is also relevant since it uses motion segmentation and tracking to learn patterns for objects and activities using visual sequence information. This is achieved by considering a codebook representation and performing hierarchical classification. A self-similarity based scheme is proposed in (Shechtman and Irani, 2007), suitable for measuring similarity between images and across video sequences. This approach introduces a novel self-similarity descriptor that computes similarity of local intensity patterns in images, and seamlessly extends to sequences of images for video matching.

This tracking intuition has also been recently applied to unsupervised classification using laser. In (Luber et al., 2008) unsupervised classification is achieved by learning exemplars built from laser scans, recovering the appearance and dynamics of objects. This approach relies on obtaining very dense scans of human-like (pedestrians and skaters) and cyclists objects, from a slow moving robot. Independent observations are considered for the tracks, and Markov models used to deal with the changing appearance over time. The work in (Schultz et al., 2003) proposes a similar scheme for tracking using exemplar models, combining in this case laser and visual information.

The fusion of laser and visual information has received considerable attention regarding intelligent transportation systems, where classification approaches have been been mostly concentrated on fully supervised techniques. In this line of research the focus is on combining the salient characteristics of the sensing modalities to perform accurate classification. The work in (Monteiro et al., 2006) performs detection, tracking and classification fusing different classifiers and combining the estimation from the various modules probabilistically. The approach in (Premebida et al., 2009) proposes alternative centralized and decentralized fusion architectures to address the classification of pedestrians. Detection and tracking of cars and pedestrians is achieved in (Spinello et al., 2009) by combining visual and laser information using implicit shape models and conditional random fields (CRF) (Ramos et al., 2007).

2.1 Contributions

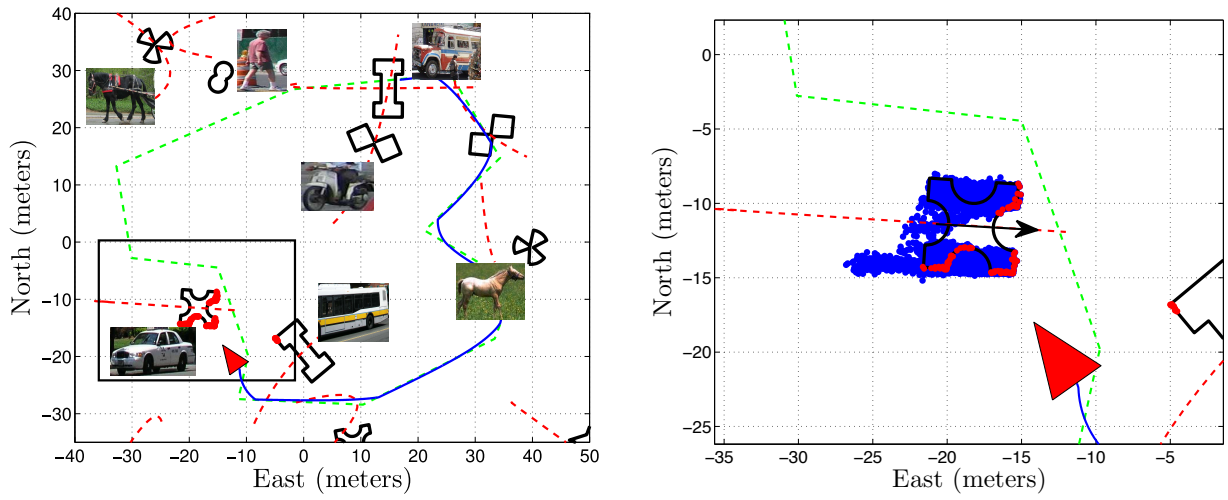
To the best of our knowledge, this work presents the first integrated architecture to perform unsupervised classification of dynamic obstacles combining laser and visual data for autonomous navigation in urban environments. The main contributions of this work are:

- The introduction of laser stamps and the formulation of a measure to compute similarity between these laser stamps. An extension of AP for performing similarity-based incremental clustering of dynamic obstacles through the associated stamp similarities.
- The representation of visual tracks through visual stamps using a single-instance feature-based approach and the formulation of a visual similarity learning method based on PL considering *a priori* clustering. A combined similarity measure is presented and an iterative clustering algorithm based on AP is derived for combined laser and visual clustering.
- The extension of the visual similarity learning approach based on the MIL framework that uses visual stamps derived from full sequences of images and better deals with challenges associated with perception in real-world scenarios.

3 Laser Stamps for Unsupervised Classification

This section explores unsupervised classification of dynamic obstacles using laser information. Temporal integration of dynamic information is addressed by combining the scans generated by tracks in the environment. The main concept behind the proposed techniques is that, by formulating an appropriate track-based representation, distinctive descriptors can be obtained to perform unsupervised classification. The clustering scheme should efficiently operate without using *ad-hoc* features or a number of clusters defined *a priori*. The architecture in this section uses the information provided by a 2D laser mounted on a moving vehicle as the input, and provides the output as the grouping associated with the sensed obstacles.

The proposed approach relies on the formulation of laser stamps and a similarity measure that is associated with these stamps. Unsupervised classification is then addressed through clustering, with focus on pairwise approaches using stamp similarities as point-to-point distances. The methodology for performing unsupervised classification of dynamic obstacles using laser can be summarized in two main stages:



(a) (b)

Figure 2: Laser tracks for stamp representation. (a) shows the vehicle (red triangle) navigating among static and dynamic obstacles. The obstacle inside the black rectangle has been detected and tracked, and the corresponding sequence of scans is extracted for stamp construction. (b) presents a zoom-in of the track in (a), showing the particles from the tracker in blue dots, and an arrow indicating the direction of movement. Note that no scan is extracted from the static object on the right. In both images, dashed green lines indicate the reference trajectory for the moving vehicle, and dashed red lines the trajectories followed by the obstacles. The simulated environment introduced here is detailed in Section 3.3. Note that there is no obvious relationship between the obstacles' contours and the images used for each of them; the associations are made only for evaluation purposes using the simulated data.

- The generation of laser stamps from the laser tracks defined by the dynamic obstacles (Section 3.1).
- The similarity-based clustering of these tracks using the derived stamp similarities (Section 3.2).

3.1 Laser Stamps for Dynamic Obstacle Representation

The approach proposed to generate laser stamps from the laser tracks defined by the dynamic obstacles, is composed of the following stages: 1) the extraction of laser tracks, 2) the computation of laser stamps and 3) the associated stamp similarity. Section 3.1.1 presents a scheme for the extraction of laser tracks that combines dynamic obstacle detection and tracking. A multi-level laser stamp representation for shape synthesis of the laser tracks is introduced in Section 3.1.2 together with the stamp similarity measure in Section 3.1.3.

3.1.1 Laser Tracks

The extraction of laser scans corresponds to the first module in Figure 1(b) of the processing pipeline, and is therefore important since the robustness of subsequent representation and clustering procedures depends on the quality of the extracted tracks.

The specific setup integrates a dynamic obstacle detection module together with a Bayesian multiple target tracker. A probabilistic laser-based motion module based on (Katz et al., 2008) first detects dynamic behavior in the laser coordinate frame given the current laser scan s_k at time k. This method implements a spatio-temporal correspondence procedure based on scan registration. Using the obtained correspondences, robust detection is performed by casting the decision problem in a probabilistic framework that considers sensor

noise and computes occlusion verification. At a high level of abstraction, this procedure can be described by the following pseudo-code function interface:

$$\left[\mathcal{Z}_k^{dyn}, \mathcal{P}_k^{dyn}\right] = \texttt{motion\_detector}(s_k), \tag{1}$$

where $\mathcal{Z}^{dyn}$ denotes the set of observations $\mathbf{z}^{dyn}$ for the detected obstacles in laser coordinates, with each $\mathbf{z}^{dyn}$ indicating the centroid of the laser segment. $\mathcal{P}^{dyn}$ denotes the set of corresponding segmented laser returns P^{dyn} for the detected obstacles. The segmentation of laser returns is performed using agglomerative hierarchical clustering in an Euclidean space (Duda et al., 2001) with a predefined distance threshold of 0.3 m.

The tracker uses the sample based joint probabilistic filter scheme by (Frank et al., 2003) performing multiple target tracking to estimate 2D position and orientation of obstacles $X_k^{dyn} = (x_k, y_k, \phi_k)$ at time k. The tracker uses a constant velocity model for the process and the observations are provided by segments obtained from the current laser scan s_k by applying clustering. Process and observation noise are assumed to be zero-mean Gaussian with known covariances; i.e., $\sigma_x = 1.5$ m, $\sigma_y = 1.5$ m, $\sigma_\phi = 0.2$ rad, $\sigma_r = 0.15$ m, $\sigma_\theta = 0.02$ rad. Tracks are initialized using information provided by the motion detection module. Each track corresponds to only one dynamic obstacle in the scene. The combined use of the tracker and the detected objects (provided by the motion detection module) allows the system to update the states and incorporate new tracks. This tracking procedure maintains the state of dynamic obstacles and provides the corresponding sequences of scans observations, and can be denoted through the following pseudo-code function:

$$\left[\mathcal{T}_k^{dyn}, \mathcal{X}_k^{dyn}\right] = \texttt{tracker}\left(\mathcal{Z}_k^{dyn}, \mathcal{P}_k^{dyn}, \mathcal{X}_{k-1}^{dyn}\right), \tag{2}$$

where $\mathcal{T}_k^{dyn}$ collects sequences of scans $\mathbf{T}_k^{dyn}$ that correspond to the set $\mathcal{X}_k^{dyn}$ of dynamic obstacles in the scene. The number of subsequent laser scans did not appear to be crucial in our experiments, as long as each tracked object contained a few scans with sufficient shape structure. There is therefore no particular constraint regarding the length of the tracks used for the representation.

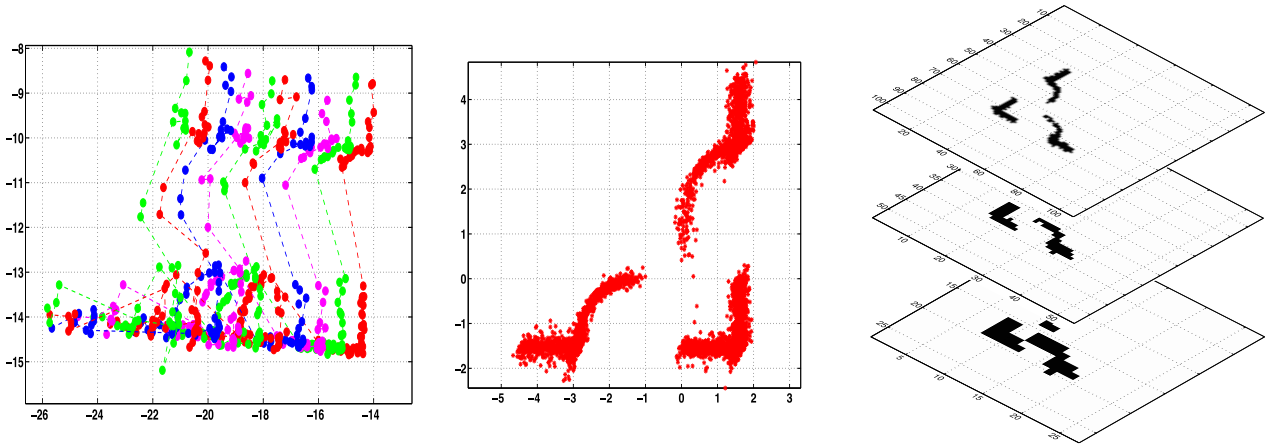
Figure 2 exemplifies the procedure to extract laser scans from tracks associated with moving obstacles. Figure 2(a) shows the simulated environment with the vehicle navigating among moving and static obstacles. One particular track is used in this scenario to illustrate the extraction of laser tracks, indicat ed in the image by the black rectangle. Figure 2(b) shows a zoom-in of the chosen track in (a). This image also includes the particles from the tracking process that correspond to this particular track.

3.1.2 Laser Stamp Representation

The aim of the method presented in this section is to obtain a robust and distinctive *stamp representation* suitable for unsupervised classification through similarity-based clustering. The stages for processing incoming laser tracks include the alignment of subsequent scans for each track, and the computation of multi-scale occupancy grids for the registered laser segments.

The term *stamp* is inspired by the modeling approach, where a distinctive mark or "signature" is used to symbolize tracks that might change over time. A *laser stamp* is defined as a pyramidal grid representation for a dynamic object, constructed by aligning subsequent laser scans from the extracted track and building a multi-level occupancy grid (Moravec and Elfes, 1985) of the registered returns. This provides a solid basis for shape synthesis that deals with the difficulties of inferring spatial appearance. The proposed scheme combines the characteristics of least-mean-square error based alignment with the advantages of occupancy grid modeling for range information. This occupancy grid is extended into a multi-scale representation for further robustness.

The proposed representation uses pyramidal levels that can be adjusted in the maps by setting different OG resolutions. This approach is closely related to the multi-scale schemes used in computer vision techniques



(a) (b) (c)

Figure 3: Laser stamp representation. (a) shows the sequence of some of the segmented scans extracted in global coordinates for the track in Figure 2, using colors to differentiate each of the scans. (b) illustrates the scans from (a) aligned and normalized (to center coordinates (0,0)) using the ICP registration procedure. (c) shows the corresponding different grids (stacked together) for each of the levels of a stamp of $L = 3$ and $\text{Res}_1 = 0.15$ m. Note that the units in (a) and (b) are meters, whereas (c) shows the corresponding grid representation in cell units rotated 90° with respect to the aligned scans in (b).

for shape synthesis sharing several important characteristics. First, the multiple levels allow the model to capture shape at different scales. Grids at the top of the pyramid tend to capture more coarse underlying features in the aligned scans, such as the spanned area (Lazebnik et al., 2006). Finer levels, on the other hand, capture smaller details in the grids. Another salient characteristic of pyramidal representations is an increased insensitivity to small rotations. As illustrated in (Bosch et al., 2007), they perform robustly even in presence of small pose variations. This is important when using aligned scans as the input, possibly subject to minor uncertainty in orientation due to noise in the tracker. Multi-scale representations provide efficient ways to compute similarity for pyramid matching (Grauman and Darrel, 2005).

Considering sequences of scans provided by laser tracks as in Figure 2, stamps are obtained through two stages: *alignment of segments*, with the input given by the segments belonging to each of the laser tracks in $\mathbf{T}_k^{dyn}$ (as provided by (2)), and the *multi-level stamp computation*, generating multi-level stamps S from the aligned segments.

Laser segments are aligned using the *iterative closest point* (ICP) registration procedure (Besl and McKay, 1992). In this work, the quaternion based method from (Horn, 1987) is used for the representation of poses and transformations, where the alignment of segments uses a local coordinate frame attached to each of the dynamic obstacles in the tracks. The employed specific variant of ICP (Rusinkiewicz and Levoy, 2001) uses constant weighting for the pairs of points and rejects the pairs which are not mutually nearest neighbors for more robust alignment.

The computation of stamps uses the set of aligned segments for each of the laser tracks. Let $\mathbf{B}^{(i)}$ denote the set of aligned segments for the track i. The set is first normalized to obtain an absolute, pose independent set $\overline{\mathbf{B}}_k^{(i)} = \{\tilde{\mathbf{b}}_{i_k}\}$. This normalization is performed with respect to location and heading given by the objects' 2D pose provided by the tracker. Using the normalized set $\overline{\mathbf{B}}_k^{(i)}$, the grid at resolution ℓ (with 2^ℓ cells) is constructed as follows. Following the notation from (Elfes, 1989), let $S^{(\ell)}(C_m)$ be the state variable that stores the probability $P(S^{(\ell)}(C_m) = Occ)$ of the cell C_m at resolution ℓ of being occupied. Since the cell states are exclusive, $P(S^{(\ell)}(C_m) = Occ) + P(S^{(\ell)}(C_m) = Free) = 1$. The evaluation of the posterior over

the occupancy of each grid cell is based on binary Bayes filters:

$$P\left(S^{(\ell)}(C_m) = Occ \mid \overline{\mathbf{B}}_k\right) \propto P\left(\tilde{\mathbf{b}}_{i_k} \mid S^{(\ell)}(C_m) = Occ\right)\ P\left(S^{(\ell)}(C_m) = Occ \mid \overline{\mathbf{B}}_{k-1}\right), \tag{3}$$

where $\overline{\mathbf{B}}_k$ represents the set of normalized observations received until time k, $P\left(S^{(\ell)}(C_m) = Occ \mid \overline{\mathbf{B}}_{k-1}\right)$ the previous estimate of the cell state and $P\left(\tilde{\mathbf{b}}_{i_k} \mid S^{(\ell)}(C_m) = Occ\right)$ is the sensor occupancy model. Assuming the laser scanner possesses independent Gaussian noise in the range and bearing readings $\mathbf{r} = [r_r, r_\theta]^T$ which generated the $\{\tilde{\mathbf{b}}_{i_k}\}$, the laser sensor model follows:

$$P(\mathbf{r} \mid \mathbf{z}) = \frac{1}{\sqrt{2\pi}\sigma_r\sigma_\theta}\exp\left[-\frac{1}{2}\left(\frac{(r_r - z_r)^2}{\sigma_r} + \frac{(r_\theta - z_\theta)^2}{\sigma_\theta}\right)\right], \tag{4}$$

where $\mathbf{z} = [z_r, z_\theta]^T$ represent the actual observations, and σ_r and σ_θ the standard deviations in range and bearing, respectively. The sensor occupancy model is then obtained from the laser sensor model applying the Kolmogorov's theorem (Elfes, 1989). In particular, this work uses the solution for the two-dimensional case presented in (Leal, 2003).

A thresholding stage is finally used to obtain a clean occupied/free stamp representation $S^{(\ell)}$, providing advantages that have been validated through extensive experimental evaluation. This thresholding is useful as a filtering mechanism, since it allows the system to deal with noisy scan segments that might be extracted from dynamic obstacles due to, for instance, pitching of the vehicle. When these scans are incorporated into the representation they are normally filtered out through the thresholding due to their low frequency of occurrence. The thresholding can be denoted as the operator $Thres(\bullet, T_{stamp})$, for a level T_{stamp}. Considering L different levels in the representation, the full multi-scale stamp is then defined as:

$$S^L = \left\{S^{(\ell)}\right\} = \left\{Thres\left(P\left(S^{(\ell)}(C_m) = Occ \mid \overline{\mathbf{B}}_k\right), T_{stamp}\right)\right\}, \tag{5}$$

for $\ell \in \{1, 2, \ldots, L\}$. A L-scale stamp will then be composed of L OG layers of different resolution. Considering an initial grid resolution Res_1 at $\ell = 1$, the L levels are computed with a spatial resolution that doubles the immediate finer one.

The alignment and multi-level computation stages for constructing laser stamps are illustrated in Figure 3 for the laser track shown in Figure 2. Figure 3(a) presents the sequence of segmented scans for the track, and Figure 3(b) shows the aligned and normalized segments using the ICP procedure. Figure 3(c) presents the computed stamp for a 3-level[1] stamp of $L = 3$ and $Res_1 = 0.15$ m. In this case, and throughout this work, the threshold level used for the stamps is $T_{stamp} = 0.5$.

3.1.3 Laser Stamp Similarity

The laser stamp representation is accompanied by an associated measure to compare and evaluate the similarity between different descriptors. Different measures can be used to compute similarity (or dissimilarity) between data points. A direct measure of similarity is the distance between data points, where various different metrics and formulations (e.g., Euclidean (Bishop, 2006) or Mahalanobis (Rousseeuw and Leroy, 1987)) can be used. Non-metric functions can also be considered providing large values when points are similar. The correlation coefficient $\rho(\mathbf{x}_i, \mathbf{x}_j) = \frac{\sigma_{ik}}{\sqrt{\sigma_i\sigma_k}}$ (Duda et al., 2001), for instance, provides values between -1 and 1, with σ_i and σ_k the variances of $\mathbf{x}_i$ and $\mathbf{x}_j$ and σ_{ik} the cross-correlation. $\rho_{ik}^2 = 1$ indicates that points are completely correlated and $\rho_{ik}^2 = 0$ if they are uncorrelated. An approach using this correlation coefficient is then preferred in this work since such a normalized measure is suitable for combination with other normalized similarities derived from probability estimates. This section formulates a *stamp similarity* measure based on correlation for the multi-level description of the tracks presented in Section 3.1.2.

[1] This work uses 3-level stamps, which provide good results empirically validated.

The formulation of stamp similarity follows the approach presented in (Grauman and Darrel, 2005) for pyramid matching. Intuitively, pyramid matching is evaluated by using a sequence of increasingly coarser grids over the feature space, and computing the distance at each level of resolution (Lazebnik et al., 2006). Let S_I^L and S_J^L be two L-scale laser stamps, as given in (5). The stamp similarity between the laser stamps S_I^L and S_J^L can be defined as:

$$sim_{laser}\left(S_I^L, S_J^L\right) = \eta \sum_{\ell=1}^{L} w_\ell \ \rho_{(\ell)}^2\left(S_I, S_J\right), \tag{6}$$

where w_ℓ is the weight at level ℓ, $\rho_{(\ell)}^2$ is the similarity between S_I and S_J at the pyramid level ℓ, and η a normalization factor. The experiments in this work use a squared normalized 2D correlation coefficient (González and Woods, 1996) for the computation of ρ as:

$$\rho_{(\ell)}\left(S_I, S_J\right) = \frac{\sum_{x,y} \overline{S}_I^{(\ell)}(x,y) \ \overline{S}_J^{(\ell)}(x,y)}{\sqrt{\sum_{x,y} \overline{S}_I^{(\ell)}(x,y)^2 \sum_{x,y} \overline{S}_J^{(\ell)}(x,y)^2}}, \tag{7}$$

where $\overline{S}^{(\ell)}(x,y) = \left[S^{(\ell)}(x,y) - \overline{S}^{(\ell)}\right]$ and $\overline{S}$ is the mean of S. Two stamps that are similar at level ℓ will be perfectly correlated, and then $\rho_{(\ell)}^2 = 1$. On the other hand, two completely different stamps will have $\rho_{(\ell)}^2 = 0$.

Because of the pyramidal structure of the stamps, the level ℓ will also consider the contributions at the finer level $\ell + 1$. The levels are then weighted using $w_\ell = \frac{1}{2^{L-\ell}}$, where ℓ is the current level, and L the total number of levels. The weight associated with each of the levels is, indeed, inversely proportional to the cell width at that level. The normalization factor is then set to $\eta = 1/\sum_{\ell=1}^{L} w_\ell$.

3.2 Similarity-based Clustering

The last stage for unsupervised classification in Figure 1(b) involves clustering laser stamps producing grouping for the tracks. Having obtained laser stamps and similarities from laser tracks in Section 3.1, the focus is now on achieving accurate unsupervised classification of tracks through efficient clustering. Considering potential clustering alternatives, affinity propagation (AP) (Frey and Dueck, 2007) is introduced to perform similarity-based clustering using stamp similarities. AP clustering seamlessly integrates within the intended unsupervised scheme, remaining independent from *ad-hoc* laser features, and without assuming a number of classes known *a priori*.

The most commonly used clustering techniques can be arranged into *central grouping* and *similarity-based clustering*. Traditional central grouping clustering uses some mixture model or parameterization of the data. Methods such as *K-means* and mixture models using *expectation-maximization* (EM) algorithm (Dempster et al., 1977) provide good quality results when the data adjusts to the pre-defined models, given a suitable features representation. These approaches have interesting geometric properties such as scatter, centroids, or exemplars (particular data points that describe clusters in a compressed way) important when modeling uncertainty or compressing the representation. In many applications, however, the data might not adjust well to the models or suitable feature descriptions might be hard to obtain. Fortunately, in many of these cases it is often possible to obtain a measure of similarity between examples, and similarity-based approaches have shown salient capabilities in a variety of related scenarios. *Hierarchical schemes* (Duda et al., 2001) and *spectral clustering* (Shi and Malik, 2000) are fully local techniques using pairs of point-to-point similarity for partitioning. One of the main disadvantages of this is the possible lack of homogeneity in the clusters. These approaches do not actually require that all the points within a cluster be similar to a single center. Moreover, they usually require the number of clusters be defined *a priori*. This issue is mitigated in AP, which achieves intercluster homogeneity performance similar to central clustering representations.

AP clustering is introduced in next subsection as a feasible solution for pairwise clustering using similarities that combines the main advantages of central grouping and similarity-based techniques. Incremental alternatives for clustering are explored in Section 3.2.2, where a novel formulation is proposed for incremental affinity propagation (AP) clustering. This incremental scheme is able to efficiently perform on-line classification and to deal with very large datasets.

3.2.1 Affinity Propagation (AP) for Clustering

Affinity propagation (AP) (Frey and Dueck, 2007; Dueck and Frey, 2007) integrates some of the advantages of model-based representations into a graph-based pairwise clustering approach. As in other pairwise techniques, AP directly examines similarity between pairs of data points. In this case, however, the clustering considers a probabilistic model to describe the data distribution. By recursively propagating affinity messages, the method is able to robustly learn a mixture model of the data, avoiding potential bad initialization issues and wrong decisions. Moreover, AP is able to consider pairwise relationships globally and identify cluster centers or exemplars. This is an important capability of the scheme, providing exemplars that synthesize and compress the information in the clusters.

The basic AP algorithm considers two types of messages derived from the following constraints. First, no cluster can be without an exemplar. Second, the clusters must contain at least one member in addition to the exemplar. Based on these constraints, the scheme works by exchanging messages between data points based on pairwise similarities, where each type considers a different kind of competition. Messages are combined to find the points that are exemplars, and the association of points with these exemplars. "Responsibility" is the message $r(i,k)$ sent from the data point i to each candidate exemplar point k. This type of message denotes how suitable a point k is as exemplar for a point i, considering other potential exemplars for i. The "availability" $a(i,k)$ message sent from candidate exemplar point k to point i accumulates evidence for how appropriate it would be for point i to choose point k as its exemplar.

Considering a dataset D with N data points, the goal is to produce a set of indices $\{c_1, c_2, \ldots, c_N\}$ indicating the index of the corresponding exemplar for the data points; i.e., if the point i is an exemplar then $c_i = i$. Let $\{sim(i,k)\}$ represent a set of real valued pairwise similarities between data points in D. The similarity $sim(i,k)$ indicates how well the data point k is an exemplar for the data point i. The number of clusters needs not to be specified *a priori*. Instead, AP takes as input a real number $sim(k,k)$ for each data point k that influences the number of clusters. These values are set to the same value if all data points are equally suitable as exemplars and can be varied to obtain different number of clusters. The median of the input similarities produces a large number of clusters, and the minimum results in a small number. Availabilities and responsibilities are combined into the *net similarity* N_S to finally identify the exemplars, where this net similarity is defined as in (Dueck, 2009) as the sum of the similarities of non-exemplar data points to their exemplars, plus the sum of the exemplar preferences (i.e., the term $(r(i,k) + a(k,i))$. AP effectively maximizes this net similarity N_S as the objective function for clustering.

3.2.2 Incremental Clustering

There are several reasons to consider incremental approaches when performing unsupervised obstacle classification through clustering. First of all, obstacle classification in robotics is an incremental process and therefore additional obstacle classes might need to be defined on-the-fly. On-line solutions often need to be produced since other modules might depend on clustering solutions obtained at running time. Timing related to the computation of very large datasets is another important factor. Two different alternatives to perform incremental clustering through pairwise grouping using similarities are examined below: *naive incremental* and *incremental clustering using exemplars*. A new approach called *Incremental AP* is then proposed which uses concepts from AP within an incremental framework.

Naive incremental clustering is a direct, "brute force" implementation of incremental clustering where the dataset is updated with new samples and clusters are recomputed each time for the fully updated dataset.

Algorithm 1: Function `exemplar()` to compress data for incremental clustering.

Input : indices $\{c_i^*\}$ of exemplars, and dataset of points D^+
Output: reduced dataset D^+, and similarity matrix $\{sim\}^+$

```
// Compress dataset using exemplars:
forall the i = c_i^* ∈ {c_i^*} do
    D^+(j)^* ← f_Exemplar (D^+, i, c_i^*, ζ)
D^+ ← D^+*
// Compress similarity:
{sim}^+ ← get_sim(D^+, D^+)
```

The approach first updates the dataset, concatenating the previous dataset D^- with the new data D^{new} into D^+. Similarity is then computed for the fully updated dataset. This incremental procedure is exact in the sense that the similarity is computed using the full updated dataset D^+. It is, however, markedly slow because of the computation of the full similarity.

The use of compressed similarities in an incremental clustering approach appears like a natural improvement to reduce similarity computation. The incremental framework proposed in (Valgren et al., 2007) addresses this compression for spectral clustering using exemplars. The key concept is to obtain representatives to replace clusters' members, in order to compress the dataset from previous iterations and reduce similarity computation. The iterative procedure aims to estimate the number of clusters on-line, a feature that is not directly provided by standard spectral clustering. If an exemplar-based clustering is utilized, the scheme naturally provides the clusters' representatives needed in the framework achieving *incremental clustering using exemplars.*

Replacing clusters' members by exemplars through a function `exemplar` as shown in Algorithm 1, greatly increases the performance of the algorithm regarding the computation of similarity. In this case, similarity computation (as indicated in Algorithm 1 through the external function `get_sim` that computes similarity between sets of data points following Equation 6) involves new data points and only exemplars from the previous iterations, which are much fewer than the original data points. A straightforward implementation includes each of the exemplars to the compressed dataset as indicated by the function $f_{Exemplar}$:

$$f_{Exemplar}\left(D^+, i, c_i^*\right) = D^+(i). \tag{8}$$

This implementation provides the maximum compression, replacing each entire cluster by its corresponding exemplar $D^+(i)$. However, since the approach uses compressed similarities, the clustering results are only approximate.

Incremental Affinity Propagation (AP) This section builds on the general framework for incremental clustering using exemplars introduced above. By exploiting the highly compressed similarities this scheme is more efficient in terms of computation at the expense of reducing the quality of the clustering. This trade-off between accuracy and computation time can be balanced by replacing each of the clusters by a set of data points instead of single exemplars. A representation using a collection of data points appears to be closer to AP clustering. Indeed, AP constitutes a pairwise clustering scheme with a global scope in the sense that the grouping incorporates the information from all the data points and not only neighbors. A set of points replacing the clusters better captures the fact that all the points, and not just the exemplars, are important for clustering. An approach for *incremental AP* clustering considering these issues is proposed below.

Effectively, the AP incremental clustering uses a modified function $f_{Exemplar}$ to compress the dataset based on the function `exemplar` from Algorithm 1. The idea is to select each of the exemplars together with a collection of additional data points to replace the clusters. Since in AP all the points in the clusters are connected to their corresponding exemplars, each collection can be visualized as a constellation of data points with an exemplar center. Denoting this function as $f_{APExemplar}$, the following definition selects exemplars

together with a number of data points for each cluster:

$$\begin{aligned} &f_{APExemplar}\left(D^{+}, i, c_i^{*}\right) = D^{+}(i) \cup Const, \\ &Const = \{D^{+}(k)\}, \quad \forall k \in c_i^{*}, \;\; k \neq i, \;\; |Const| = K_{AP}, \end{aligned} \tag{9}$$

where $Const = \{D^{+}(k)\}$ includes K_{AP} neighbors of the $D^{+}(i)$ exemplars to the compressed datasets. The number of neighbors K_{AP} provides a tuning mechanism for the `exemplar` algorithm. K_{AP} can select very dense constellations for light compression (thus slow computation) and high accuracy, or fewer points for strong compression and less accurate but faster computation. This parameter links in the extremes the two approaches presented before. Although K_{AP} can be set to a fixed value, information provided by the AP clustering process can be integrated for an automatic selection for each cluster.

An automatic selection of neighbors is proposed based on information provided by the AP algorithm. Recalling from Section 3.2.1, AP maximizes the net similarity N_S as the objective function, where N_S is defined as the total sum of the similarities of non-exemplar data points to their exemplars. A similar concept can be specified for each of the clusters with respect to their exemplars. For each cluster j, the *cluster net similarity* N_{S_j} is defined as the net similarity restricted to the cluster, i.e., the sum of similarities of non-exemplar data points to the exemplar in the cluster j. Larger N_{S_j} indicates that the cluster j better fits in the complete dataset of points. Conversely, lower N_{S_j} means the cluster is not well represented. This concept of cluster net similarity can be used to find the appropriate number of neighbors to effectively represent each cluster. Considering negative similarities with a maximum value for net similarity of zero[2], the number of neighbors $K_{AP}^{(j)}$ for each cluster j can be defined as:

$$K_{AP}^{(j)} = \frac{N_{S_j}}{N_S} \;\; N, \tag{10}$$

where N is the total number of data points. This equation provides a simple mechanism to select the number of neighbors for each cluster, where few points are selected if the cluster net similarity is high, and more are needed for clusters with low net similarity. The number of neighbors $K_{AP}^{(j)}$ should be consistent with the actual size of each cluster, and can be further limited to maintain a bounded complexity:

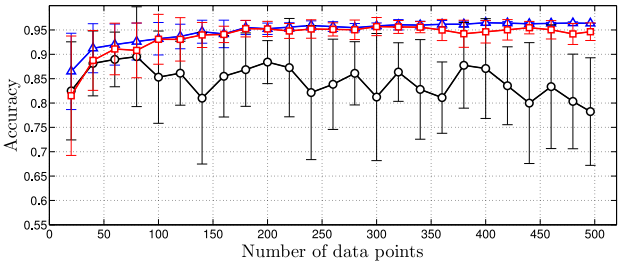
$$K_{AP}^{(j)} = \min\left(\frac{N_{S_j}}{N_S} \;\; N, N_j, K_{Max}\right), \tag{11}$$

where N_j is the number of data points in cluster j, and K_{Max} is the maximum number of neighbors. It is important to note that although clusters are replaced by exemplars and neighbors and are not further used in the AP incremental clustering, they can actually be maintained for future use.

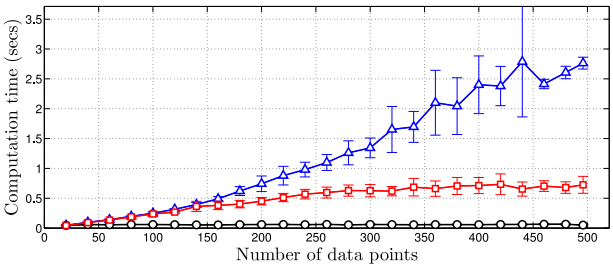
3.3 Simulation

The performance of unsupervised classification using laser stamps is demonstrated here using the simulated environment shown in Figure 2(a). This simulator produces both laser and visual data[3] sensed from a vehicle navigating in an environment populated with static and dynamic obstacles, in an area of approximately 100 by 100 meters. Nine different possible obstacle classes 1 to 9 are employed which consider various distinct synthetic shapes (shown using solid black in Figure 2(a)). Dynamic obstacles follow randomly generated reference trajectories with changing speeds. During navigation, observations are logged using the laser returns sensed by the moving vehicle on the surrounding obstacles. Images are associated with each of the obstacles from a database obtained from the public labeled LabelMe dataset (Russell et al., 2008). These images are randomly selected from a subset of segmented images which are obtained by querying for labels in the dataset with the following 'class:label' associations: 1:bike, 2:pedestrian, 3:car, 4:bus, 5:truck, 6:animal, 7:dog, 8:boat, and 9:other.

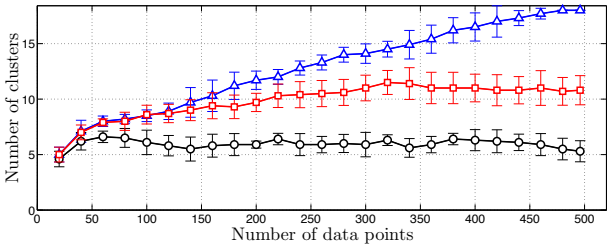
[2]Note, however, that most similarity measures can be made negative through simple operations.
[3]Only laser data is used in this section, visual information is considered in Section 4.3.



(a) Accuracy versus number of data points



(b) Computation time versus number of data points



(c) Number of clusters versus number of data points

Figure 4: Performance of incremental clustering. Accuracy, computation time, and number of clusters for the incremental methods versus the number of data points. In the plots, blue '△', black '○', and red '□' curves correspond to naive, incremental using only the exemplars and AP incremental (K_{max} = 20) clustering methods, respectively. Solid lines represent the mean performance, and error bars indicate standard deviations.

Table 1: Confusion matrix using AP incremental clustering. Rows denote real (ground truth) classes and columns the AP classification results. The accuracy (in %) and the obtained number of clusters are shown at the bottom.

	Class 1	Class 2	Class 3
Class 1 (bike)	167	0	0
Class 2 (pedestrian)	0	155	16
Class 3 (car)	0	8	125
Accuracy (%) = 95.34 Number of clusters = 10			

The evaluation presented here considers a three-class scenario which includes a labeled dataset of 471 tracks with a number of tracks per class (1/2/3) of 167/163/141 extracted using the procedure presented in Section 3.1.1. Laser stamps are then computed from the laser tracks following Section 3.1.2, with pairwise similarities computed as presented in Section 3.1.3. The performance of the AP incremental clustering approach presented in Section 3.2.2 is validated for the three-class scenario, considering the three incremental methods: naive incremental, incremental using only the exemplars, and AP incremental clustering. Incremental clustering is evaluated for each of the three methods considering fixed segments of size $\mathcal{L}$ drawn from the dataset. Effectively, these segments of $\mathcal{L} = 20$ data points are randomly extracted from the total labeled dataset and sequentially added to the previous data to produce an incremental clustering scenario. This process is repeated 100 times, randomly shuffling the dataset each time such that 100 different random sequences are considered. The evaluation criterion used for the algorithms throughout this work is both the accuracy and the number of clusters, and here computation time is also reported for completeness. Mean and standard deviations are obtained for accuracy, computation time and number of clusters for each of the methods with respect to the number of data points. The results of this experiment are shown in Figure 4.

As can be seen in the plots, the naive incremental approach (blue '△' curve) is the most accurate of the three, reaching an accuracy of 96.37% at the end of the sequence. The computation in this case takes 2.77 secs ($\sigma = 0.1$), and detects 18 clusters. Incremental using only the exemplars clustering (black '∘' curve) is much faster (0.0506 secs with $\sigma = 0.01$) but its accuracy is below 80%. AP incremental clustering (red '□' curve) reaches 94.65% ($\sigma = 1.81$) in 0.72 secs ($\sigma = 0.14$) while detecting 10.8 clusters ($\sigma = 1.31$) using functions implemented in Matlab and running on a 2.33 GHz processor. The performance of this AP incremental approach is comparable to the performance obtained using the naive incremental clustering. AP incremental reaches the accuracy of the naive approach after roughly 100 data points, with considerably lower computation times and fewer detected clusters. Table 1 shows results for one instance[4] selected from this experiment, with an accuracy of 95.34% and finding 10 clusters.

Additional experiments were undertaken (not included in this paper for space reasons) considering a nine-class scenario achieving similar results. The experimentation shows that the laser-based architecture is very accurate for unsupervised classification. It also indicates that using laser information only, the system tends to over-cluster the data and generates a large number of clusters. This issue is addressed in further sections by incorporating vision into the clustering process.

4 Integrating Vision for Unsupervised Classification

The results derived from the classification of laser stamps above suggest that, although the method is very accurate, it tends to over-cluster. An excessive number of generated clusters might not be suitable for many applications that make use of the classification outcome. This section builds on the results of similarity-based classification using laser data and integrates visual information in order to improve the clustering for unsupervised classification of dynamic obstacles. The algorithms aim to improve the models associated with dynamic obstacles using visual information[5] and a given initial clustering, as produced by the unsupervised classification of laser tracks from Section 3.

The proposed architecture uses images provided by a color monocular camera as the input, where each of the images is associated with each of the laser tracks. The clustering results obtained using laser are also considered as inputs. The output of the system is the grouping produced for the sensed dynamic obstacles. The incorporation of vision into the process follows a structure similar to the sequence used for processing laser information. The methodology for combining visual and laser modalities contains two main stages:

- The processing of visual tracks to obtain visual stamps (Section 4.1).
- The computation of a combined laser and visual similarity for similarity-based clustering (Section 4.2).

4.1 Visual Stamps for Dynamic Obstacle Representation

This section presents a visual representation that is suitable for combining with laser tracks. Throughout this section it is assumed that one image containing the dynamic object can be extracted for each track and used in the visual representation. This assumption simplifies the presentation of concepts and algorithms for visual integration, providing a framework that is extended in Section 5 for complete sequences of images. The proposed algorithm is composed of the following stages: 1) the computation of visual stamps and 2) their corresponding similarity.

[4] Experiments in Section 4 build on this particular instance and results when dealing with combined clustering in a three-class scenario.

[5] The emphasis of the unsupervised algorithms presented hereafter is on schemes able to learn accurate models in times suitable for retraining but not necessarily in real-time.

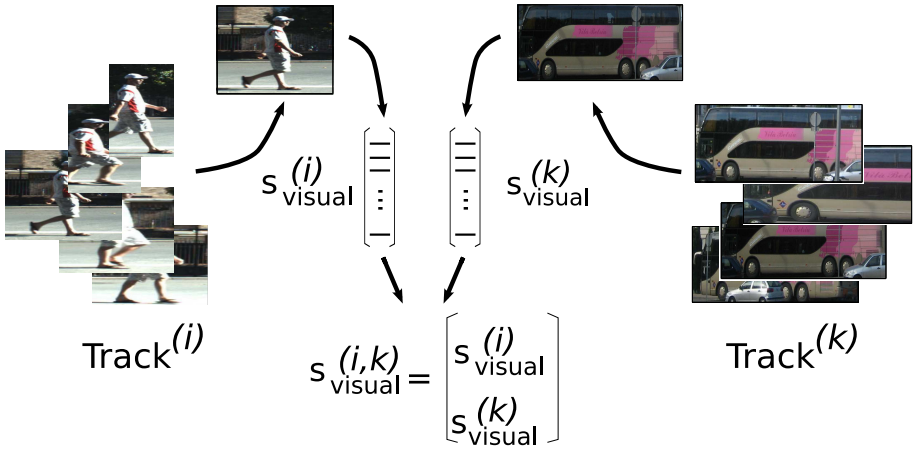


Figure 5: Visual stamps using a single-instance representation. The visual stamp $S^{(i)}_{visual}$ for each track i is modeled using a feature vector $\mathbf{f}^{(i)}_{visual}$ extracted from one of its images. For the computation of similarity between two tracks i and k, a combined visual stamp $S^{(i,k)}_{visual}$ is arranged by stacking up the individual stamps.

4.1.1 Visual Stamp Representation

This section introduces a single-instance feature-based approach for the visual stamps that utilizes only one image per track that can be chosen, for instance, by random selection out of the entire track. This "single-instance" term follows the standard framework of discriminative supervised learning (Duda et al., 2001), where data points are represented by feature vectors from some data space $\mathcal{X}$ of d dimensions, i.e., $\mathcal{X} = \mathbb{R}^d$. The goal here is to learn a classifier function f, such that $f : \mathcal{X} \rightarrow \mathcal{Y}$, where $\mathcal{Y} = \{-1, 1\}$ (or equivalently $\mathcal{Y} = \{0, 1\}$) are labels indicating the classes for binary classification. A training dataset of N pairs $\{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_N, y_N)\}$ is used in the training stage to learn the function f, where $\mathbf{x}_i \in \mathcal{X}$ and $y_i \in \mathcal{Y}$. This classifier function f can then be used to predict the labels for new data points in a posterior inference stage.

Although labels are not explicitly mentioned at this stage, the terminology is convenient since it refers to one image out of several possible images in the track. It is assumed that each laser track generates a sequence of *region of interest* (ROI) in images that can be obtained through calibration (described in Section 6.1), and one of these images is chosen from the sequence containing the object of interest.

The process to obtain single-instance visual stamps and combined visual stamps from visual tracks is illustrated in the diagram in Figure 5. Each track i is described by a *visual stamp* following a feature representation as:

$$S^{(i)}_{visual} = \mathbf{f}^{(i)}_{visual}, \tag{12}$$

where $\mathbf{f}^{(i)}_{visual} \in \mathbb{R}^d$ is the feature vector, and d is the number of visual features extracted from the representative image. This work uses a dense representation of features for the images (Dalal, 2006), where different image features are densely extracted over each entire image, and then stacked together into a single high-dimensional feature vector $\mathbf{f}^{(i)}_{visual}$. This approach allows the combination of various types of robust features and has been extensively used for object detection (Viola and Jones, 2004; Lowe, 2004). For the comparison of two tracks i and k, a *combined visual stamp* is arranged by stacking up the individual stamps:

$$S^{(i,k)}_{visual} = \begin{bmatrix} S^{(i)}_{visual} \\ S^{(k)}_{visual} \end{bmatrix}. \tag{13}$$

This representation allows to obtain the similarity function in a discriminative supervised learning framework.

4.1.2 Visual Stamp Similarity

This section presents a method to learn visual similarity. Rather than constructing an explicit function to directly compute distance as in Section 3.1.3 for laser data, a learning approach is used to estimate visual similarity. An initial process providing an accurate but over-clustered *a priori* grouping of tracks (i.e., the laser-based results as presented in Section 3) determines a *positive-only learning* (PL) scenario. Examples included in the same clusters share common characteristics since they belong to the same class. These examples can be considered similar and constitute the positive training set in a supervised scheme. No assumption can be made regarding examples between different groups which depend on the (unknown) underlying classes of the groups and therefore define unlabeled training data.

For the sake of generality, the formulation of PL learning follows standard learning notation introduced in Section 4.1.1. Let $\mathbf{x}$ denote a data point or example, and $y \in \{0, 1\}$ be a binary label. Let l be an additional random variable such that $l = 1$ if the example $\mathbf{x}$ is labeled, and let $l = 0$ if $\mathbf{x}$ is unlabeled. For the positive-only scenario, labels are certain only when $l = 1$, that is $y = 1$ (positive examples). $l = 0$ indicates that the examples are unlabeled, then either $y = 1$ or $y = 0$ may be true. PL learning estimates a continuous classifier function $f(\mathbf{x})$ such that $f(\mathbf{x}) = p(y = 1|\mathbf{x})$, given positive and unlabeled examples grouped in triples in $P = \{(\mathbf{x}_p, y_p, l_p)\}$ and $U = \{(\mathbf{x}_u, y_u, l_u)\}$ respectively.

There are two main approaches to learn a classifier in a positive-only scenario. The more common one is to identify examples likely to be negative and then apply a standard learning method to these and to the positive examples (Yu, 2005). The other approach is to weight the unlabeled examples and then train a classifier using these and the positive examples (Elkan and Noto, 2008). This latter approach is used in this work, since it provides a general framework for learning that includes a mechanism for choosing different weights for the different unlabeled examples. Section 4.1.2 formulates the visual similarity learning algorithm using the positive-only (PL) learning framework. As in (Elkan and Noto, 2008), the experiments in this work use the SVM library from (Chang and Lin, 2008) (extension for "weighted instances") with probability output estimates as described in (Wu et al., 2003). Experiments have been undertaken using weighted instances for boosting (Freund and Schapire, 2006) achieving equivalent performance in most of the cases.

Visual Similarity Positive-only Learning This section formulates the visual similarity learning algorithm using the positive-only learning (PL) framework. The proposed scheme follows the traditional *training-inference* (Duda et al., 2001) procedure. By exploiting *a priori* clustering, positive and unlabeled training data can be extracted from each pair of clusters to learn similarity models using the PL framework. Once similarity models are obtained for each pair of clusters, similarity measures *sim* are inferred by computing the corresponding similarity function. This process is presented below.

Let $\{c_{ini}\} = \{c_1, c_2, \ldots, c_n\}$ indicate the initial clustering, and $sim^{(i,k)}_{visual}$ the visual similarity to be estimated for the pair (c_i, c_k), with $c_i, c_k \in c_{ini}$. Appropriate positive and unlabeled P and U training sets can be generated by considering the pairs of clusters of interest. Effectively, for the pair of clusters (c_i, c_k), the positive set is $P = \{(\mathbf{x}_p, y_p, l_p)\}$, where:

$$\begin{aligned} \mathbf{x}_p = &\left\{ S^{(m,n)}_{visual} \right\}, \quad \forall\, S^{(m)}_{visual}, S^{(n)}_{visual} \quad \text{such as} \\ &m \in c_i,\, n \in c_i,\, m \neq n, \quad \text{or} \\ &m \in c_k,\, n \in c_k,\, m \neq n; \end{aligned} \tag{14}$$

with $y_p = 1$ and $l_p = 1$, for all elements p in P. This includes all the possible combinations for the visual stamps in each cluster i and k. The unlabeled set is $U = \{(\mathbf{x}_u, y_u, l_u)\}$, where:

$$\begin{aligned} \mathbf{x}_u = &\left\{ S^{(m,n)}_{visual} \right\}, \quad \forall\, S^{(m)}_{visual}, S^{(n)}_{visual} \quad \text{such as} \\ &m \in c_i,\, n \in \{c_{ini} \backslash \{c_i \cup c_k\}\}, \quad \text{or} \\ &m \in c_k,\, n \in \{c_{ini} \backslash \{c_i \cup c_k\}\}; \end{aligned} \tag{15}$$

with $l_u = 0$ for all elements u in U (unknown y_u). This comprises all the possible combinations for the visual stamps between each cluster i and k and the rest of the clusters. For both P and U, $S^{(m,n)}_{visual}$ follows (13) for combined stamps.

Considering P and U, the visual similarity model $f(\mathbf{x})$ for the pair of clusters (c_i, c_k) is obtained using the algorithm from (Elkan and Noto, 2008). The actual similarity $sim^{(i,k)}_{visual}$ is then evaluated by instantiating the learned model:

$$\left\{sim^{(i,k)}_{visual}\right\} = f\left(\mathbf{x}_t\right), \tag{16}$$

for the pair of clusters (c_i, c_k), where $\mathbf{x}_t$ is the inference set:

$$\begin{aligned}\mathbf{x}_t = \left\{S^{(m,n)}_{visual}\right\}; \quad \forall\, S^{(m)}_{visual}, S^{(n)}_{visual} \quad \text{such as} \\ m \in c_i,\, n \in c_k,\end{aligned} \tag{17}$$

which includes all the combinations of visual stamps between the clusters i and k. In this manner, $\left\{sim^{(i,k)}_{visual}\right\}$ generates both discrete classification (i.e., $\{\hat{y}_t\} \in \{0, 1\}$ indicating 'dissimilar' and 'similar') and probability estimates of similarity for each of the data points in $\mathbf{x}_t$.

The complete process described in this section comprising (14)-(17) can be summarized through the following function:

$$\left\{sim^{(i,k)}_{visual}\right\} = \texttt{learn\_VisualSimil}\left(\{c_{ini}\}, S_{visual}\right), \tag{18}$$

which obtains the visual similarity for all given data points with respect to all pairs of clusters given the clustering $\{c_{ini}\}$.

4.2 Laser and Vision Similarity-based Clustering

Having presented a visual representation and an approach to learn visual similarity in Section 4.1, this section formulates the integration of laser and vision for similarity-based clustering.

4.2.1 Combined Similarity

The approach for combining laser and visual similarities is based on a linear combination. Each separate similarity is weighted based on information derived from the visual learning process, considering the level of certainty achieved in the estimation process.

Let $sim^{(i,k)}_{comb}$ denote the combined similarity for the clusters i and k, for a given initial clustering $\{c_i\}$. The combined similarity is defined as:

$$sim^{(i,k)}_{comb} = w^{(i,k)}_{visual} \,.\, \overline{sim^{(i,k)}_{visual}} + w^{(i,k)}_{laser} \,.\, sim^{(i,k)}_{laser}, \tag{19}$$

where $\overline{sim^{(i,k)}_{visual}}$ derives from the visual similarity $\left\{sim^{(i,k)}_{visual}\right\}$ obtained from (18), and $sim^{(i,k)}_{laser}$ is the laser similarity that produced the clustering $\{c_i\}$. The weights $w^{(i,k)}_{visual}$ and $w^{(i,k)}_{laser}$ denote the importance given to the visual and laser similarities in the combination. The higher the value for a particular weight, the more the combined similarity relies on this individual contribution. The visual similarity method described in Section 4.1.2 provides probability estimates of similarity $sim^{(i,k)}_{visual}$ for all data points between pairs of clusters. In this case, a higher visual similarity between the clusters i and k will "link" these clusters stronger through the combined similarity. The inclusion of the components $\overline{sim^{(i,k)}_{visual}}$ and $w^{(i,k)}_{visual}$ accounts for the underlying uncertainty of the visual similarity estimation process, as described below.

The approach to deal with this uncertainty uses the discrete classification set $\{\hat{y}_t\}$ obtained from Section 4.1.2, associated with the continuous estimated similarities. Let T_{pos} denote the $N^{(i,k)}_{similar}$ data points classified as

Algorithm 2: Iterative combined clustering.

Input : initial clustering $\{c_{ini}\}$, set of visual stamps S_{visual}, set of laser similarities $\{sim_{laser}\}$
Output: indices $\{c_i\}$ indicating clusters for data points

`// Initialization:`
1 $\{c_i\} = \{c_{ini}\}$, $\{sim_{comb}\} = \{sim_{laser}\}$
`// Iterative clustering:`
2 **while** *not converged* **do**
  `// Learn visual similarity:`
3 $\left\{sim_{visual}^{(i,k)}\right\} \leftarrow$ `learn_VisualSimil`$(\{c_i\}, S_{visual})$
  `// Combined similarity update:`
4 **forall the** $i, k \in c_i$ **do**
5 $w^{(i,k)} \leftarrow \frac{N_{similar}^{(i,k)}}{N_{total}^{(i,k)}}$
6 $sim_{comb}^{(i,k)} \leftarrow w^{(i,k)} \,.\, sim_{visual}^{(i,k)} + (1 - w^{(i,k)}) \,.\, sim_{comb}^{(i,k)}$
  `// Reclustering:`
7 $\{c_i\} \leftarrow$ `cluster`$(\{sim_{comb}\})$

'similar' in the discrete classification for visual similarity, i.e., $\hat{y}_j = 1$ for all $j \in T_{pos}$. The term $\overline{sim_{visual}^{(i,k)}}$ is defined as the mean visual similarity over the members of T_{pos} as:

$$\overline{sim_{visual}^{(i,k)}} = \left(\frac{1}{N_{similar}^{(i,k)}}\right) \sum_{j \in T_{pos}} sim_{visual}^{(i,k)}\Big|_j. \tag{20}$$

The weight $w_{visual}^{(i,k)}$ associated with the visual similarity is defined as:

$$w_{visual}^{(i,k)} = \frac{N_{similar}^{(i,k)}}{N_{total}^{(i,k)}}, \tag{21}$$

where $N_{total}^{(i,k)}$ is the total number of data points for the clusters i and k. The weight $w_{laser}^{(i,k)}$ for the laser similarity is defined as the complement of the weight for visual similarity, i.e., $w_{laser}^{(i,k)} = 1 - w_{visual}^{(i,k)}$. The parameters defined in (20)-(21) utilize only data points classified as 'similar'. This provides an adaptive contribution of the individual similarities, tuned with respect to the certainty (or uncertainty) obtained from the visual estimation process.

4.2.2 Iterative Combined Clustering

This section incorporates the combined visual and laser similarity formulated above into an iterative scheme for similarity-based clustering. The proposed framework allows the refinement of the clustering by means of iteratively learning the visual similarity, computing the combined similarity, and reclustering. The algorithm for iterative combined clustering is shown in Algorithm 2.

Considering an initial clustering $\{c_{ini}\}$ generated by a set of laser similarities sim_{laser}, and a set of visual stamps S_{visual}, the goal is to refine the initial clustering producing a more accurate clustering $\{c_i\}$. The algorithm is initialized in step 1, setting the current clustering $\{c_i\}$ and combined similarity $\{sim_{comb}\}$ to the input values, $\{c_{ini}\}$ and $\{sim_{laser}\}$ respectively. Then, it iteratively repeats three main operations in steps 2-7 until a convergence criterion is reached. The convergence criterion used in this work is a given number of iterations with no change in clustering. Visual similarity is first learned in step 3 using the approach denoted by (18) and the current clustering. In steps 4-6, the combined similarity is updated following (19), considering the laser similarity and the current estimation of visual similarity. Finally, the combined similarity is used in step 7, where clustering is recomputed using the AP similarity-based clustering.

Table 2: Set of visual features for the visual stamp representation.

Type	Feature	Dimen.
Shape	Straight lines	4
	Canny	6
Color	RGB histogram	27
	RGB min/max/index min/index max/channel	5
	HSV histogram	27
	HSV min/max/index min/index max/channel	5
	Color gradient histogram	36
	Intensity and entropy histogram	21
Texture	Steerable pyramid histogram	609
	Steerable pyramid min/max	2
	SIFT, # descriptors, # descriptors normal.	130
	Haar features	24
	MSER	4
	PHOG	40

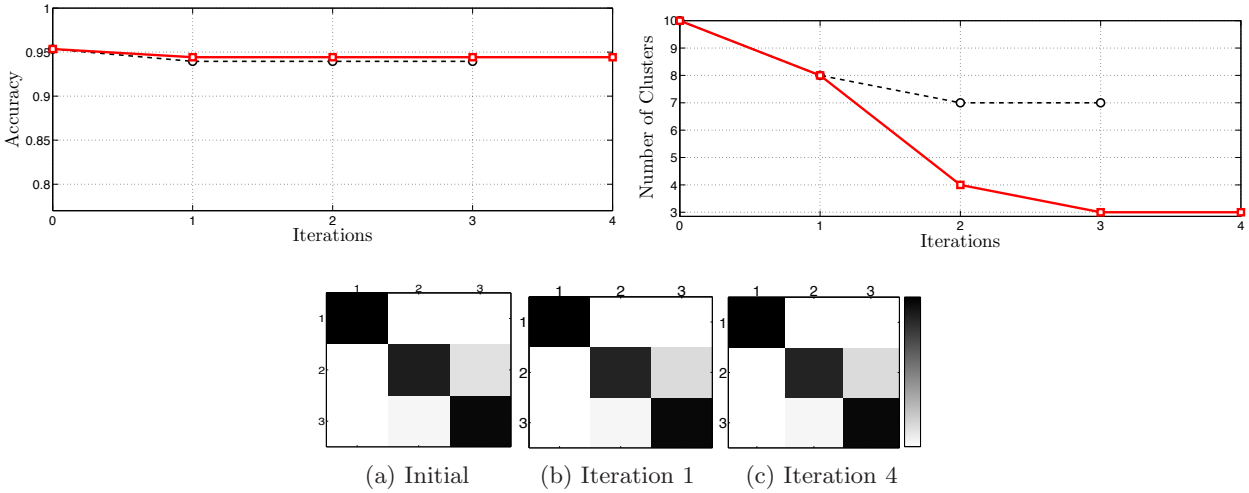
4.3 Simulation

The performance of the clustering scheme for combined clustering using laser and vision is evaluated for the three-class scenario. The dataset introduced in Section 3.3 is utilized in this case considering one image associated to each of the tracks. Extracted tracks now provide laser stamps, together with associated images that correspond to the underlying class. Visual stamps are computed for each of the images associated with the tracks following (12), such that data points $\left\{S_{laser}^{(i)}, S_{visual}^{(i)}\right\}$ contain both laser stamps $S_{laser}^{(i)}$ and visual stamps $S_{visual}^{(i)}$.

In this evaluation a standard set of features is utilized for visual stamp representation. The features extracted for each of the images include a variety of shape, color and texture features, with a total of 14 different types as shown in Table 2. Shape is captured computing straight line features (i.e., # lines, # lines (normalized), length of longest line, feature indicating if longest line is vertical), and Canny edges (i.e., # features, # features (normalized), # straight lines in Canny edges, # straight lines in Canny edges (normalized), length of longest straight line in Canny edges, feature indicating if longest straight line in Canny edges is vertical). Color features comprise 3D histograms of the RGB and HSV channels, minimum and maximum values for the channels, color gradient histograms, and intensity and entropy histograms. Texture features include steerable pyramid (Simoncelli and Freeman, 1995) coefficients of the image, the minimum and maximum steerable pyramid coefficients, SIFT features (Lowe, 2004), Haar features (Viola and Jones, 2004), maximally stable extremal regions (MSER) (Matas et al., 2002), and pyramid histograms of oriented gradients (PHOG) (Bosch et al., 2007). By concatenating all these features, feature vectors $\mathbf{f}_{visual}$ of dimension $d = 940$ are obtained for each track.

Algorithm 2 is used for iterative combined clustering, with laser similarities $\{sim_{laser}\}$ provided by (6). An initial clustering $\{c_{ini}\}$ is utilized for starting the iterative procedure, given by the laser stamps and indicated by the results shown earlier in Table 1. As shown in the table, the accuracy obtained through laser stamps only is 95.34% with 10 obtained clusters. The goal of the iterative process that includes visual information is to refine this clustering, reducing the number of clusters while maintaining high accuracy. Figure 6 shows results of the iterative combined clustering process for this three-class scenario.

The procedure converges in four iterations, as shown in solid red '□' plots in the top left and right images of Figure 6 for the clustering accuracy and number of clusters respectively. As can be seen, the number of clusters is substantially reduced from 10 to 3 clusters with no significant reduction in accuracy (95.34% to



(a) Initial (b) Iteration 1 (c) Iteration 4

Figure 6: Iterative combined clustering. The top left and right images show in solid red '□' the evolution of clustering accuracy and the number of clusters with respect to the number of iterations. Results for the standard visual similarity estimation are also included for comparison in dashed black '○'. Bottom images (a)-(c) present the confusion matrices obtained at different iterations.

94.4%). The use of a standard classifier that assumes unlabeled data points as negative is also included for comparison, i.e., an SVM classifier using positive training data points and considering unlabeled as negative data points. Dashed black '○' plots indicate the evolution of this simplified approach into the iterative scheme, which converges early in three iterations after reducing the number of clusters from 10 to 7. The bottom images (a)-(c) show a visual representation of the confusion matrices obtained at different iterations. Gray level in each cell indicates accuracy, with white cells representing zero and black indicating maximum accuracy. The confusion matrices show strong diagonal, with minimum hits in the off-diagonal terms that would indicate incorrect classification.

The experiments show that the system is able to maintain high accuracy for combined laser and vision unsupervised classification while notably reducing the number of clusters. For the presented three-class scenario, the accuracy was maintained over 94% and 3 clusters that corresponded to the obstacle classes were found from an initial estimation of 10 custers produced by the laser data only. The experimentation shows that if the visual similarity estimation process is able to learn accurate models then the system converges without major oscillations. This is due to the fact that through the combined similarity (together with the weights) the regrouping tends to reduce the number of clusters from an initial accurate *a priori* clustering.

5 Extended Visual Stamps for Dynamic Obstacle Representation

The visual model proposed above is restrictive regarding sequences of images derived from demanding real-world visual tracks. Visual tracks obtained using real platforms can be affected by anomalies that are detrimental to the classification process, including slight sensing errors, occlusion, errors in the calibration and timing of the sensors, projection artifacts and errors such as misalignment of the ROIs in the images with respect to the dynamic obstacles. The extraction of training examples from this challenging scenario is a difficult task. Variations in appearance can affect the visual representation where models are obtained from possibly incorrect observations.

The scenario defined by visual tracks presents different representation alternatives. A single-instance representation (as in Section 4) can be used assuming that one image can be extracted to robustly synthesize

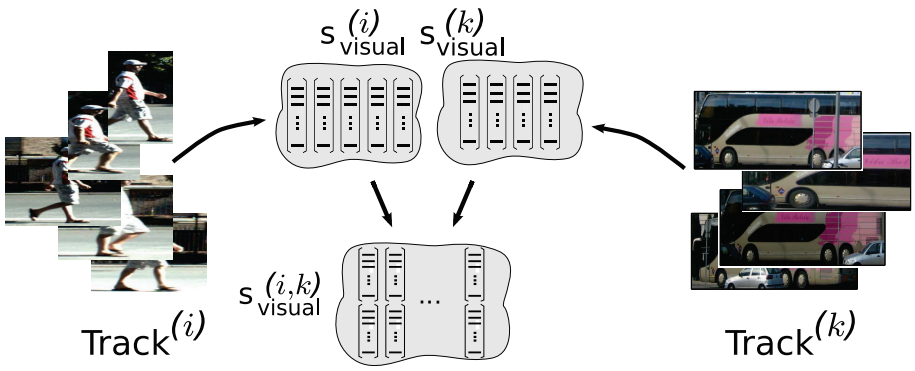


Figure 7: Visual stamps using a multiple instance MIL representation. The MIL visual stamp $S^{(i)}_{visual}$ for each track i is modeled using a bag of feature vectors $\mathbf{f}^{(i)}_{visual\,j}$ extracted from each of the images. For the computation of similarity between two tracks i and k, a combined MIL visual stamp $S^{(i,k)}_{visual}$ is arranged by stacking up pairs of individual feature vectors from both bags.

the appearance of the dynamic object in the entire sequence. Due to the inaccuracies described earlier, this simplification can potentially select suboptimal training examples. More robust representations with multiple training examples can also be used, utilizing full sequences and without assuming any representative image for the tracks. Data point instances $\mathbf{x}_i$ generated for the images can be regarded as independent visual stamps for traditional single-instance learning. This can lead to poor performance (Babenko et al., 2008) in the learning process, confusing the classifiers by assuming that all instances are significant. Another option is to consider a *bag* $\mathbf{X}$ containing several data point instances, each of these obtained from the images in the track. Each bag can then be used as the visual stamp for the track in a *multiple instance learning* (MIL) (Long and Tan, 1996; Dietterich et al., 1997) representation. This MIL representation nicely captures the structure of the problem, where bags of data points instances are obtained from tracks. MIL is used in this section to represent extended visual stamps, providing a versatile framework that is suitable for learning.

5.1 Multiple Instance (MIL) Framework

Supervised learning for sets can be formulated following (Dietterich et al., 1997). In this case, data is represented by bags $\mathbf{X}_i = \{\mathbf{x}_{i1}, \ldots, \mathbf{x}_{ib}\}$ of data points $\mathbf{x}_{ij}$ from some space $\mathcal{X} = \mathbb{R}^d$ such that $\mathbf{x}_{ij} \in \mathcal{X}$. The data space for the bags is denoted by $\mathcal{X}^b$, where b is the cardinality for the bags assumed fixed for notational simplicity. The goal is to learn a classifier function $\mathcal{F} : \mathcal{X}^b \rightarrow \mathcal{Y}$, where $\mathcal{Y} = \{-1, 1\}$ (or equivalently, $\mathcal{Y} = \{0, 1\}$) for binary classification. A training dataset of N pairs $\{(\mathbf{X}_1, y_1), (\mathbf{X}_2, y_2), \ldots, (\mathbf{X}_N, y_N)\}$ is used in the training stage to learn the function $\mathcal{F}$, where each $\mathbf{X}_i \in \mathcal{X}^b$ and $y_i \in \mathcal{Y}$ is a bag label. This classifier function is then used to predict the labels for new bags in an inference stage. In MIL (Dietterich et al., 1997), a bag label is positive if at least one of its instances is positive, that is:

$$y_i = \max_j(y_{ij}), \tag{22}$$

where y_{ij} are the instance labels ($y_{ij} \in \mathcal{Y}$) that are assumed to exist but are not known during training. Several algorithms have been proposed to solve the MIL problem, most involving a generalization of existing algorithms to the multiple instance formulation.

5.2 MIL Visual Stamp Representation

This section presents the formulation of the extended MIL visual stamps. A MIL representation is used now for the tracks, where MIL visual stamps and combined MIL stamps are derived. The process for obtaining these MIL visual stamps and combined MIL visual stamps is shown in Figure 7.

Each track i can be described by a MIL *visual stamp* as a bag of feature vectors:

$$S^{(i)}_{visual} = \left\{ \mathbf{f}^{(i)}_{visual\,1}, \ldots, \mathbf{f}^{(i)}_{visual\,b} \right\}, \tag{23}$$

where $\mathbf{f}^{(i)}_{visual\,j} \in \mathbb{R}^d$ is a feature vector extracted from each of the b images in the track as in Section 4.1.1. For the comparison of two tracks i and k, a MIL *combined visual stamp* can be arranged stacking up pairs of individual feature vectors from the individual MIL stamps:

$$\begin{gathered} S^{(i,k)}_{visual} = \left\{ \begin{bmatrix} \mathbf{f}^{(i)}_{visual\,m} \\ \mathbf{f}^{(k)}_{visual\,n} \end{bmatrix} \right\}; \quad \forall\, \mathbf{f}^{(i)}_{visual\,m}, \mathbf{f}^{(k)}_{visual\,n} \quad \text{such as} \\ \mathbf{f}^{(i)}_{visual\,m} \in S^{(i)}_{visual}, \quad \mathbf{f}^{(k)}_{visual\,n} \in S^{(k)}_{visual}. \end{gathered} \tag{24}$$

(23)-(24) define bags of feature vectors consistent with the MIL framework introduced above in Section 5.1. In summary, bags are used to represent visual tracks as extended MIL visual stamps, and then arranged into MIL combined visual stamps for computing similarity.

5.3 MIL Visual Stamp Similarity

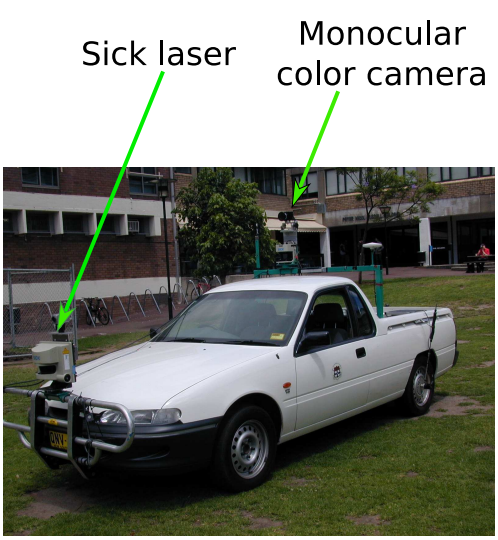
The procedure for obtaining visual similarity described in Section 4.1.2 is originally formulated based on single-instance visual stamps. This section details how this can be adapted to deal with the multiple instance model. Two issues need to be considered to perform this upgrade: (i) the MIL stamp representation formulated in (23)-(24) should be used as the new visual stamps for the tracks, and (ii) suitable MIL classifiers need to be integrated into the visual similarity learning algorithms.

The pseudo-function in (18) summarizes the procedure to obtain visual similarity given single-instance visual stamps S_{visual} and an initial clustering $\{c_{ini}\}$. Visual stamps are effectively used within this process when visual similarity models are obtained for pairs of clusters using a positive-only learning approach (i.e., using (Elkan and Noto, 2008)). This algorithm can be adapted to support MIL stamps by using MIL classifiers instead of standard classifiers. In this work, this is achieved using the MILBoost (Viola et al., 2005) classifier, with prior distribution indicating weighted bags instead of weighted instances. MILBoost derives the multiple instance variant of boosting (Freund and Schapire, 2006). The weak classifiers used within the boosting were implemented using decision stumps. These stumps perform weak classication by defining thresholds along each of the dimensions of the feature space. The learning process achieved good convergence at 40 iterations.

The iterative combined clustering algorithm presented in Section 4.2.2 uses (18) (see Algorithm 2, step 3) to iteratively learn visual similarity and improve clustering. By considering the adapted procedure to learn visual similarity using MIL stamps described above, the iterative process is suited to handle the extended visual stamps.

6 Results

This section shows experimental results of the track-based classification algorithm in urban scenarios. As in Section 4, the goal is to integrate visual information with laser data to obtain accurate classification by clustering with a reduced number of clusters representing the underlying obstacle classes. Visual tracks obtained from real platforms (such as the experimental vehicle shown in Figure 8(a)) present very demanding classification scenarios with perception issues that affect the classification process. To cope with these challenges the extended visual stamps for track-based classification introduced in Section 5 are considered here for the visual models.



(a) Experimental vehicle

(b) Data in urban scenarios

Figure 8: Image (a) presents the experimental vehicle used for data collection in urban environments. A 2D Sick laser and a high-resolution monocular camera are mounted at the front and on the roof of the vehicle respectively. Images in (b) show different examples of the registered laser and vision data provided by the experimental platform.

6.1 Experimental Platform

The sensor data used in the experiments comprises 2D laser and visual information. The considered platform consists of the experimental vehicle shown in Figure 8(a). This research vehicle is equipped with a 2D laser at the front mounted at a height of 1 m, and a high-resolution monocular color camera on the roof. The 2D laser is a SICK LMS291 (Sick, 2005) and the camera is an iDS UI2250C (iDS, 2008) with resolution of 1280x1024. These sensors are integrated through a PC104 machine running real-time QNX (QNX, 2005). The system provided by (ACFR et al., 2006) is used to log the data. Images are provided by the iDS camera at 5 Hz, and laser scans are logged using the 2D Sick laser in high speed mode at 75 Hz. Collected datasets consist of 60 minutes of data of sequences of images temporally correlated with laser scans. In the experiments, the speed of the experimental platform varies between 0-40 km/h through city-like campus roads and streets in and around the University of Sydney campus.

The projection of the laser returns onto the camera images is achieved by estimating the transformation from the laser to the image coordinate frame using the camera/laser calibration procedure described in (Zhang and Pless, 2004). In this approach, estimates for the extrinsic transformation parameters are obtained by performing constrained non-linear optimization using training data. This data is obtained by positioning a planar calibration pattern (a checkerboard) in various poses. Registration information is also used to consider ROIs on the images. Clusters of laser returns are used to determine ROIs taking into account the mean range to the objects in the laser coordinate frame. Examples of the registered laser and vision data together with the ROIs for some of the obstacles are shown in Figure 8(b).

6.2 Experiments

The objectives of the experiments are (i) to demonstrate the performance of the track-based algorithms for unsupervised classification in urban scenarios combining laser and vision, and (ii) to compare the extended MIL visual stamp representation introduced in Section 5 with respect to the single-instance approach used

Table 3: Confusion matrix using AP clustering. The accuracy (in %) and the obtained number of clusters are shown at the bottom.

	Class 1	Class 2	Class 3
Class 1 (bike)	25	3	0
Class 2 (pedestrian)	6	113	3
Class 3 (car)	2	0	59
Accuracy (%) = 93.37 Number of clusters = 8			

in Section 4. The experiments utilize data collected using the research experimental vehicle described in Section 6.1.

The data collected in urban environments includes three main obstacle classes: 1:bike, 2:pedestrian and 3:car; and therefore the datasets are manually labeled according to these classes for ground thruth. The total number of processed tracks is 211, with a number of obstacles per class (1/2/3) of 33/116/62. The extracted laser tracks also determine visual tracks through sensor registration. Complete sequences of images contained in the visual tracks are now used to obtain extended visual stamps, using the multiple instance (MIL) framework presented in Section 5. MIL visual stamps are computed for the tracks following (23), such that data points $\left\{S_{laser}^{(i)}, S_{visual}^{(i)}\right\}$ contain both laser stamps $S_{laser}^{(i)}$ and extended visual stamps $S_{visual}^{(i)}$. The set of features used in the experiments is the set presented in Section 4.3 (detailed in Table 2). By concatenating all these features, feature vectors $\mathbf{f}_{visual\,j}^{(i)}$ of dimension $d = 940$ are obtained for each of the instances included in the bags representing the tracks. Combined MIL visual stamps are computed following (24).

Following the processing sequence from Figure 1, classification using laser is first performed, providing an initial clustering on which visual tracks build upon. Then, visual information is incorporated considering extended visual stamps in order to refine the classification results. Finally, the performance of the single-instance visual stamp approach from Section 4 is compared with the MIL extended visual stamp representation from Section 5 using real-world data.

6.2.1 Track-based Classification using Laser

This section describes unsupervised classification using laser stamps only. Laser stamp similarity sim_{laser} is computed for the laser stamps $\left\{S_{laser}^{(i)}\right\}$ of dataset described above following the procedure presented in Section 3.1.3 (see (6)). AP clustering is then applied using the Matlab MEX code from (Frey and Dueck, 2007), classifying the tracks in an unsupervised manner as presented in Section 3.2.1. Ground truth labels are only used to compute classification accuracy. The obtained results are very accurate, reaching 93.37% with 8 clusters. Table 3 presents these classification results through the corresponding confusion matrix. Table 4 shows the same AP clustering results with the ground truth class distribution (classes 1, 2, and 3) indicated for each of the obtained clusters. Classes corresponding to the exemplars (Exemplar Class) in the clusters are also shown. The total number of exemplars associated with each of the classes in this experiment is indicated in the last row in Table 4. As can be seen, class 1 (bike) is entirely captured by one exemplar, while the other two classes are represented by few exemplars that capture various views of the obstacles. The performance of the laser-based scheme was not affected by the speed of the objects; the system was able to incorporate tracks to the database for classification as long as they were consistently detected and tracked by the laser scanner operating in high-speed mode.

Table 4: AP clustering results. Rows contain the real (ground truth) classes for each of the obtained clusters indicated in the first column. The last column shows the class that corresponds to the exemplar in each cluster. The last row shows the total number of exemplars identified for each of the classes.

Cluster	Class 1	Class 2	Class 3	Exemplar
1	3	10	0	2
2	0	0	10	3
3	2	22	2	2
4	1	38	1	2
5	0	43	0	2
6	2	0	25	3
7	25	3	0	1
8	0	0	24	3
Ex. per class	1	4	3	

6.2.2 Track-based Classification Integrating Vision

Visual information is now used to improve clustering by integrating extended visual stamps into the clustering process, using the iterative combined clustering scheme from Section 4.2.2 extended in Section 5. The accuracy obtained through unsupervised classification using laser stamps only is 93.37% with 8 obtained clusters. The goal of the iterative process is to incorporate visual information to refine the clustering.

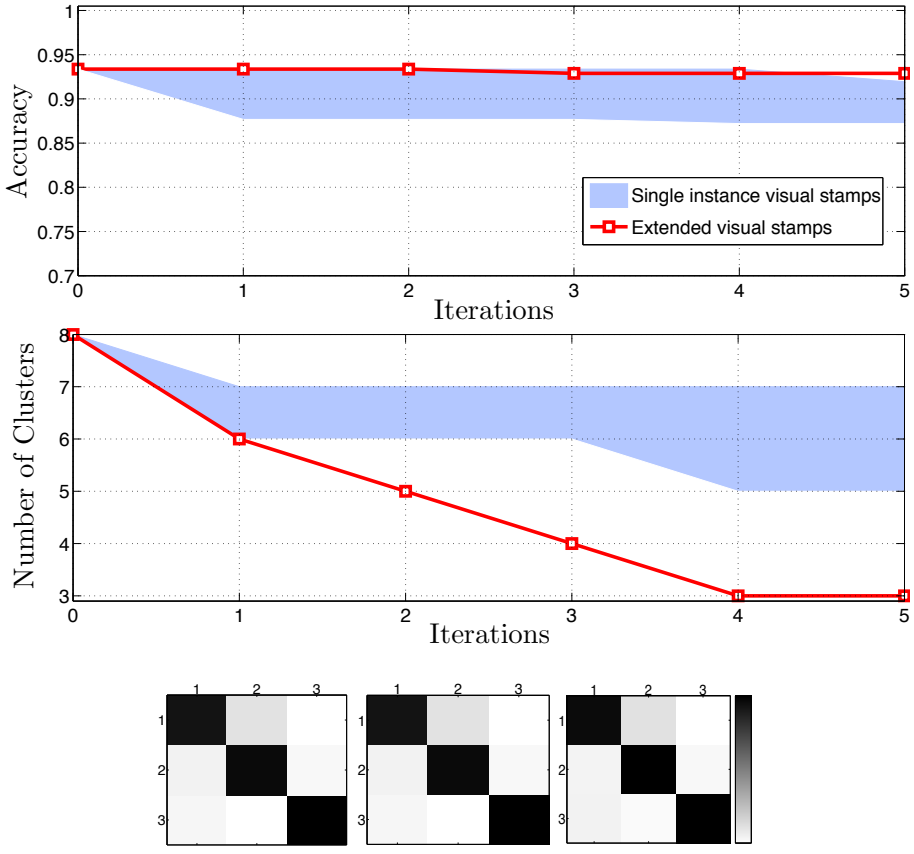
Iterative combined clustering is evaluated using laser stamps $S^{(i)}_{laser}$ and extended visual stamps $S^{(i)}_{visual}$. Laser similarities $\{sim_{laser}\}$ are obtained from laser stamps, providing the initial laser clustering $\{c_{ini}\}$ shown in Tables 3-4. As described in Section 5, extended visual stamps are combined with laser for iterative combined clustering using Algorithm 2. Convergence of the iterative scheme is achieved when the reclustering stops changing the grouping. Figure 9 shows results of this iterative combined clustering process.

The procedure converges in five iterations, as shown in solid red '□' plots in the top and center images of Figure 9 for clustering accuracy and number of clusters, respectively. The number of clusters is markedly reduced from 8 to 3 clusters, with no significant reduction of the accuracy (93.37% to 92.9%). The bottom images (a)-(c) show a visual representation of the confusion matrices obtained at different iterations. The confusion matrices show strong diagonal, with minimum hits in the off-diagonal terms. The total computation time of the iterative process for convergence is 18 minutes running unoptimized Matlab code on a 2.33 GHz processor, suggesting that the scheme is suitable for on-line retraining. The obtained computation time indicates that by providing a vehicle with the proposed architecture, the system could regularly retrain the models of the surrounding dynamic obstacles to achieve an adaptive behavior for long term navigation tasks.

6.2.3 Single-instance versus Extended Visual Stamps

The single-instance approach used in Section 4 is compared here with the MIL extended visual stamp representation introduced in Section 5. The goal is to evaluate the performance of the simplified visual scheme when dealing with more demanding visual tracks in urban scenarios, provided in this case by the dataset collected using the experimental vehicle. Using this data the experiment below computes iterative combined clustering using laser and single-instance visual stamps.

Recalling from Section 4.1, one image is extracted from each track to obtain the single-instance visual stamp representation. The procedure for the evaluation randomly selects individual images from the complete sequences of images for each of the visual tracks. For each of the tracks, the selected image is used to compute a single-instance visual stamp that is then used into the iterative combined clustering process in the same manner as in Section 4.2.2. The same process is evaluated 100 times with different seeds and computed until convergence for each case. Results of this experiment are overlaid in the top and center



(a) Initial (b) Iteration 2 (c) Iteration 5

Figure 9: Iterative combined clustering in track-based classification. The top and center images plot in solid red '□' the evolution of clustering accuracy and the number of clusters (with respect to the number of iterations) when using the MIL extended visual stamps. (a)-(c) present the confusion matrices obtained at different iterations. Light blue areas in top and center plots indicate iterative combined clustering performance when using single-instance visual stamps as detailed in Section 6.2.3.

images in Figure 9 together with the results previously obtained using the MIL extended visual tracks for comparison. The light blue areas indicate the evolution of the 100 iterative clustering processes when using single-instance visual stamps, and solid red '□' curves show the performance when considering the extended visual stamps already presented in Figure 9.

As can be seen, the use of single-instance visual stamps notably deteriorates the performance in this demanding urban dataset. Clustering quality is reduced with worst cases reaching accuracies of 87%. Moreover, the iterative procedure is not capable of minimizing the number of clusters with a maximum refinement that obtains 5 clusters. When using the MIL extended visual stamps, on the other hand, the iterative combined process is able to maintain high accuracy and reduce the number of clusters to the correct number that corresponds to the obstacle classes in the data.

7 Discussion

The proposed algorithm for track-based unsupervised classification using 2D laser and vision deploys a hierarchical formulation regarding the way sensor data is globally processed. The sensor hierarchy provides the means to decouple the sensor modalities to address problems separately and achieve unsupervised classifica-

tion. This "laser then vision" scheme omits, however, the multi-modal nature of the sensor data that could be considered and benefit some of the low-level processing stages. It also explains the reasons behind the classification failures, i.e., errors in the initial clustering produced by the laser are carried on to the second stage where vision is incorporated. This second stage uses the results from the *a priori* clustering and is then unable to correct the misclassified situations (as can be seen in Figure 9).

The emphasis on this work is on the development of unsupervised algorithms capable of learning accurate models. The experiments showed high performance regarding accuracy and computation times that suggests that on-line retraining for ITS is feasible. For example, the retraining may be needed in a scenario where the ITS moves from a city like environment to a farm. The concept of on-line introduced in the paper is from the point of view of the application rather than the algorithms. We believe that the vehicle can use the first few minutes to gradually update the models (retraining) without need to do this instantaneously. The presented scheme does not perform real-time classification, since this was considered beyond the scope of this work. The algorithms were implemented using unoptimized Matlab code, and therefore the reported computation times could be considerably reduced utilizing optimized C/C++ code. The fact that the proposed architecture obtains high-quality unsupervised classification would permit the potential incorporation of supervised stages for real-time computation.

8 Conclusions

This work developed solutions to the problem of unsupervised classification of dynamic obstacles by introducing a track-based model for the integration of laser and visual information. Regarding the processing of laser tracks, this work contributed a representation called laser stamps and a similarity measure, together with an incremental approach for AP clustering to produce efficient clustering of laser stamps for on-line clustering. With respect to visual information, visual stamps were introduced to describe visual tracks using a single-instance feature-based formulation for images representative of the entire tracks. A method based on PL learning was introduced to compute visual similarity building on results obtained from the laser stamp formulation. The visual similarity measure was then combined with the laser similarity and integrated into an iterative combined clustering scheme. Finally, extended visual models were proposed by exploiting full sequences of images from the visual tracks to better cope with challenging real-world scenarios. MIL was introduced to deal with the multiple visual stamps that are derived from visual tracks and used to extend the visual similarity learning approach.

The whole architecture was successfully validated through experiments for track-based unsupervised classification using data collected in urban environments. The experiments demonstrated the high-quality performance of the scheme, with accuracy of over 92% and finding the 3 clusters that corresponded to the obstacle classes in the data. Moreover, the system was able to robustly refine an initial clustering given by the laser while maintaining accurate grouping. It was also shown that the MIL extended visual stamp representation has improved performance regarding unsupervised classification when using real-world visual tracks obtained in urban scenarios.

The proposed architecture achieves good results in unsupervised classification of dynamic obstacles. There is, however, room for improvement with various possible extensions that can be considered. A natural extension of this work is to use 3D laser sensing instead of 2D, for example utilizing data provided by a Velodyne sensor (Velodyne, 2008). The advantages of 3D data (e..g., more dense scans, invariance against nodding of the vehicle and wider field of view) could improve the processing stages at the expense of an increase in the associated cost. A 3D Velodyne sensor is approximately ten times more expensive than a standard 2D Sick laser and this is a limiting factor for its application in the context of automotive applications. Multi-layer laser scanners, such as the Ibeo Lux (Ibeo, 2009), could potentially establish an interesting trade-off both between 2D and full 3D representations and costs. The use of RADAR (Jansson, 2005) for obstacle detection is widely spread in commercial vehicles due to reliability and cost and its use to obtain the initial classification estimates (instead of laser sensors) is worth exploring.

This work assumes a hierarchical flow in the processing of the sensor data establishing precedence of laser over vision. This occurs, for instance, in the stages performing motion detection and object tracking as well as in the aligning procedure used to produce laser stamps. Although the performance of these modules is satisfactory, "simultaneous" laser and vision processing could have been considered to further boost the efficiency. Detection and tracking in the laser coordinate frame and also in the image (Sun et al., 2006) could be used to increase range and resolution and to better deal with occlusions. Alignment of laser scans could also benefit from vision, integrating visual features as in CRF-matching (Ramos et al., 2007). This could potentially provide more robust association for creating the stamp representation.

Acknowledgment

This work is supported by the Australian Research Council (ARC) Centre of Excellence program and the New South Wales Government.

References

ACFR, U. of Sydney and LCR, U. N. del Sur. (2006). PAATV/UTE Projects. Technical Report.

Babenko, B., Dollár, P., Tu, Z., and Belongie, S. (2008). Simultaneous Learning and Alignment: Multi-Instance and Multi-Pose Learning. Technical Report, University of California at San Diego (UCSD).

Besl, P. and McKay, N. (1992). A Method for Registration of 3-D Shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer.

Bosch, A., Zisserman, A., and Munoz, X. (2007). Representing Shape with a Spatial Pyramid Kernel. In *CIVR '07: Proceedings of the* 6^{th} *ACM International Conference on Image and Video Retrieval*, pages 401–408, Amsterdam, The Netherlands. ACM.

Burl, M., Weber, M., and Perona, P. (1998). A Probabilistic Approach to Object Recognition using Local Photometry and Global Geometry. In *Proceedings of the* 5^{th} *European Conference on Computer Vision*, Freiburg, Germany.

Chang, C. and Lin, C. (2008). A Library for Support Vector Machines. http://www.csie.ntu.edu.tw/∼cjlin/libsvm/.

Dalal, N. (2006). Finding People in Images and Videos. PhD thesis, Institut National Polytechnique de Grenoble.

DARPA, D. A. R. P. A. (2007). DARPA Urban Challenge. http://www.darpa.mil/grandchallenge/.

Dempster, A., Laird, N. M., and Rubin, D. B. (1977). Maximum Likelihood from Incomplete Data Via the EM Algorithm. *Journal of the Royal Statistical Society*, 39(1):1–38.

Dietterich, T. G., Lathrop, R. H., and Lozano-Perez, T. (1997). Solving the Multiple Instance Problem with Axis-parallel Rectangles. *Artificial Intelligence*, 89(1-2):31–71.

Duda, R., Hart, P., and Stork, D. (2001). *Pattern Classification*. John-Wiley, New York.

Dueck, D. (2009). Affinity Propagation: Clustering Data by Passing Messages. PhD thesis, Department of Electrical and Computer Engineering, University of Toronto, Canada.

Dueck, D. and Frey, B. (2007). Non-metric Affinity Propagation for Unsupervised Image Categorization. In *ICCV '07: Proceedings of the 2007 ICCV International Conference on Computer Vision*, Rio de Janeiro, Brazil. IEEE Computer Society.

Elfes, A. (1989). Occupancy Grids: A Probabilistic Framework for Robot Perception and Navigation. PhD thesis, Department of Electrical and Computer Engineering, Carnegie Mellon University.

Elkan, C. and Noto, K. (2008). Learning Classifiers from Only Positive and Unlabeled Data. In *Knowledge Discovery and Data Mining KDD 2008*, Las Vegas, Nevada. ACM.

Fischler, M. A. and Elschlager, R. A. (1973). The Representation and Matching of Pictorial Structures. *IEEE Transactions on Computers*, C-22(1):67–92.

Frank, O., Nieto, J., Guivant, J., and Scheding, S. (2003). Multiple Target Tracking using Sequential Monte Carlo Methods and Statistical Data Association. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Las Vegas, USA. IEEE.

Freund, Y. and Schapire, R. E. (2006). Experiments with a New Boosting Algorithm. In *Thirteenth International Conference on Machine Learning*, pages 148–156. Morgan Kaufmann.

Frey, B. and Dueck, D. (2007). Clustering by Passing Messages Between Data Points. *Science*, 315:972–976.

González, R. and Woods, R. (1996). *Digital Image Processing*. Addison-Wesley, Wilmington, USA.

Grauman, K. and Darrel, T. (2005). Pyramid Match Kernels: Discriminative Classification with Sets of Image Features. In *ICCV '05: Proceedings of the 2005 ICCV International Conference on Computer Vision*, San Diego CA. IEEE Computer Society.

Horn, B. (1987). Closed-form Solution of Absolute Orientation using Unit Quaternions. *Journal of the Optical Society of America A*, 4(4):629–642.

Ibeo (2009). Ibeo Lux. http://www.ibeo-as.com/english/products_ibeolux.asp.

iDS (2008). I. D. Systems. http://www.ids-imaging.com/.

Jansson, J. (2005). Collision Avoidance Theory with Application to Automotive Collision Mitigation. PhD thesis, Department of Electrical Engineering, Linköping University, Sweden.

Katz, R., Nieto, J., and Nebot, E. (2008). Probabilistic Scheme for Laser Based Motion Detection. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 161–166, Nice, France. IEEE.

Lazebnik, S., Schmid, C., and Ponce, C. (2006). Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories. In *CVPR '06: Proceedings of the 2006 Conference on Computer Vision and Pattern Recognition (CVPR '06)*, New York, NY. IEEE Computer Society.

Leal, J. (2003). Stochastic Environment Representation. PhD thesis, University of Sydney, Australia.

Leordeanu, M. and Collins, R. (2005). Unsupervised Learning of Object Features from Video Sequences. In *CVPR '05: Proceedings of the 2005 Conference on Computer Vision and Pattern Recognition (CVPR '05)*, San Diego CA. IEEE Computer Society.

Long, P. M. and Tan, L. (1996). PAC Learning Axis Aligned Rectangles with Respect to Product Distributions from Multiple-instances Examples. In *Proceedings Comp. Learning Theory*.

Lowe, D. (2004). Discriminative Image Features from Scale-invariant Keypoints. *International Journal of Computer Vision*, 60(2):91–110.

Luber, M., Arras, K. O., Plagemann, C., and Burgard, W. (2008). Classifying Dynamic Objects: An Unsupervised Learning Approach. In *Robotics: Science and Systems*. MIT Press.

Matas, J., Chum, O., Urban, M., and Pajdla, T. (2002). Robust Wide Baseline Stereo from Maximally Stable Extremal Regions. In *British Machine Vision Conference 2002*. British Machine Vision Association.

Monteiro, G., Premebida, C., Peixoto, P., and Nunes, U. (2006). Tracking and Classification of Dynamic Obstacles Using Laser Range Finder and Vision. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Beijing, China. IEEE.

Moravec, H. and Elfes, A. (1985). High Resolution Maps from Wide Angle Sonar. In *International Conference on Robotics and Automation*. IEEE.

NHTSA, National Highway Traffic Safety Administration (2008). Traffic Safety Facts 2008. U.S. Department of Transportation.

Petersson, L., Fletcher, L., Zelinsky, A., Barnes, N., and Arnell, F. (2006). Towards Safer Roads by Integration of Road Scene Monitoring and Vehicle Control. *The International Journal of Robotics Research*, 25(1):53–72.

Premebida, C., Ludwig, O., and Nunes, U. (2009). LIDAR and Vision-based Pedestrian Detection System. *Journal of Field Robotics*, 26(9):696–711.

QNX (2005). QNX Programmer's Guide. http://www.qnx.com/.

Ramanan, D. and Forsyth, D. (1999). Using Temporal Coherence to Build Models of Animals. In *ICCV '99: Proceedings of the 1999 ICCV International Conference on Computer Vision*, Corfu, Greece. IEEE.

Ramos, F., Fox, D., and Durrant-Whyte, H. (2007). CRF-matching: Conditional Random Fields for Feature-based Scan Matching. In *Robotics: Science and Systems*. MIT Press.

Rousseeuw, P. and Leroy, A. (1987). *Robust Regression and Outlier Detection*. New York: Wiley.

Rusinkiewicz, S. and Levoy, M. (2001). Efficient Variants of the ICP Algorithm. In *Third International Conference on 3D Digital Imaging and Modeling (3DIM)*.

Russell, B., Torralba, A., Murphy, K., and Freeman, W. T. (2008). LabelMe: a Database and Web-based Tool for Image Annotation. *International Journal of Computer Vision*, 77(1-3):157–173.

Schultz, D., Burgard, W., Fox, D., and Cremers, A. (2003). People Tracking with a Mobile Robot Using Sample-based Joint Probabilistic Data Association Filters. *The International Journal of Robotics Research*, 22(2):99–116.

Shechtman, E. and Irani, M. (2007). Matching Local Self-Similarities across Images and Videos. In *CVPR '07: Proceedings of the 2007 Conference on Computer Vision and Pattern Recognition (CVPR '07)*, Minneapolis, MN. IEEE Computer Society.

Shi, J. and Malik, J. (2000). Normalized Cuts and Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905.

Sick (2005). Sick Laser Data Sheets. http://www.sick.de/.

Simoncelli, E. and Freeman, W. (1995). The Steerable Pyramid: A Flexible Architecture for Multi-Scale Derivative Computation. *International Conference on Image Processing*, 3:444–447.

Spinello, L., Triebel, R., and Siegwart, R. (2009). Multiclass Multimodal Detection and Tracking in Urban Environments . In 7^{th} *International Conference on Field and Service Robotics*, Cambridge, Massachusetts.

Stauffer, C. and Grimson, W. E. L. (2000). Learning Patterns of Activity Using Real-Time Tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):747–757.

Sun, Z., Bebis, G., and Miller, R. (2006). On-Road Vehicle Detection: A Review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(5):694–711.

Valgren, C., Duckett, T., and Lilienthal, A. (2007). Incremental Spectral Clustering and its Application to Topological Mapping. In *International Conference on Robotics and Automation*, Rome, Italy. IEEE.

Velodyne (2008). High Definition Lidar. http://www.velodyne.com/lidar/.

Viola, P. and Jones, M. (2004). Robust Real-time Object Detection. *International Journal of Computer Vision*, 57:2.

Viola, P., Platt, J., and Zhang, C. (2005). Multiple Instance Boosting for Object Detection. In *NIPS: Advances in Neural Information Processing Systems 19*.

Weber, M., Welling, M., and Perona, P. (2000). Towards Automatic Discovery of Categories. In *CVPR '00: Proceedings of the 2000 Conference on Computer Vision and Pattern Recognition (CVPR '00)*, Hilton Head Island, South Carolina. IEEE Computer Society.

Worral, S. (2009). Providing Situation Awareness in Complex Multi-Vehicle Operations. PhD thesis, University of Sydney, Australia.

Wu, T., Lin, C., and Weng, R. C. (2003). Probability Estimates for Multi-class Classification by Pairwise Coupling. *Journal of Machine Learning Research*, 5:975–1005.

Yamamura, Y., Tabeand, M., Murakami, T., and Kanehira, M. (2001). Development of an Adaptive Cruise Control System with Stop-and-go Capability. In *Society of Automotive Engineers 2001 World Congress*, Detroit, MI. SAE.

Yu, H. (2005). Single-Class Classification with Mapping Convergence. *Machine Learning*, 61(1-3):49–69.

Zhang, Q. and Pless, R. (2004). Extrinsic Calibration for a Camera and Laser Ranger Finder (Improves Camera Intrinsic Calibration). In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Japan. IEEE.