2010 IEEE International Conference on Robotics and Automation
Anchorage Convention District
May 3-8, 2010, Anchorage, Alaska, USA

Inferring the Semantics of Direction Signs in Public Places

Jérôme Maye*, Luciano Spinello*†, Rudolph Triebel*, and Roland Siegwart*
* Autonomous Systems Lab, ETH Zurich, Switzerland
email: {jerome.maye, rudolph.triebel, roland.siegwart}@mavt.ethz.ch
† Social Robotics Lab, Department of Computer Science, University of Freiburg, Germany
email: spinello@informatik.uni-freiburg.de

Abstract— **Most large-scale public environments provide direction signs to facilitate the orientation for humans and to find their way to a goal location in the environment. Thus, for a robot operating in the same environment, it would be beneficial to interpret such signs correctly for a safe and efficient navigation.**

In this work, we propose a novel approach to infer the meaning of direction signs and to use that for navigation, i.e., to find a mapping of a detected sign to a motion direction. Our method uses a hierarchical extension of the Implicit Shape Model framework called HISM that does not require any hand-labeled training data to detect the signs. On the lower level of this two-stage hierarchy, ISM is applied to image descriptors as in the standard approach. On the higher level, ISM operates on subparts of signs called tokens, using weights learned from data. The interpretation of the signs is inferred by associating navigation data to direction instructions. We conducted experiments from image data acquired in an airport terminal, aiming towards the implementation of a robotic guide, with promising results.

I. Introduction

Human beings can relatively easily find their way in an unknown environment using direction signs. The first use of these artifacts goes back to the Roman Empire, where milestones were placed along the dense road network to indicate the distances to the nearby major cities. Since that time, signs evolved in a more convenient form which generally consists of a symbol suggesting the direction, a distance indicator, and a part identifying the destination. Signs are nowadays not only restricted to roads. They actually serve as the main cues for navigation in most public places like train stations, airport terminals, or event venues. Given the high density of direction signs in our daily environment, it would thus be beneficial for a robot to be able to read these signs correctly for an efficient navigation. This paper explores this idea and applies it to a typical environment where the signs are of particular importance: an airport terminal.

Our approach addresses this challenge by reasoning on single images. The method presented in this paper is not based on models that are manually designed beforehand, but instead infers them from data. Moreover, we design a system that is able to generalize over several categories of signs by using object detection techniques. Whenever a sign is presented to the system, it is divided into subparts or *tokens*. The sign is therefore characterized by a hierarchical method which builds on a hierarchy of Implicit Shape Models (ISM) [1]. Specifically, a sign is defined by a geometrical arrangement of subparts and these subparts are again defined by a geometrical arrangement of primitive image feature descriptors. We term this method *Hierarchical Implicit Shape Models (HISM)*. Furthermore, if several kinds of signs are presented to the robot, this method allows to understand which are the most distinctive subparts by using a smart weighting approach. For further learning, we use our sign detection technique to find a mapping between detections and motion directions, thus inferring the meaning of direction arrows. This is achieved by analyzing the frequency of certain subparts related to signs.

In this paper, we show possible applications for a robotic guide that navigates in an unknown environment or for a robotic assistant that identifies which sign to follow to reach a desired goal.

In particular, the major contributions of this work are:

- Hierarchical Implicit Shape Models (HISM): a hierarchical subdivision and voting strategy of a sign. This allows robustness and subpart weighting.
- Unsupervised subpart clustering and description as an object: uniformly clustered color regions are described by a geometrical voting model of standard image features.
- Mapping actions to detections to learn semantical sign information: unsupervised detection of direction arrows.

The rest of the paper is organized as follows. Section II reviews the related work in the domain. Section III describes our approach for learning signs. Section IV shows how we achieve sign detection. Section V demonstrates how to map actions to sign detections. Section VI presents experiments. Section VII outlines our conclusions and future work.

II. Related Work

To our knowledge, there has been little work in the topic of unsupervised sign analysis. The work of Quingji *et al.* [2] is based on a similar experimental environment. It makes use of a Pan-Tilt-Zoom camera for obtaining detection based on SIFT features [3] without any further reasoning.

Other literature focuses on traffic signs: either their detection, their recognition, or both. For detection, the most interesting approaches are inspired by the object detection method proposed by [4], which is based on a cascade of boosted classifiers working on Haar-like features. Several authors report promising results with this technique [5], [6], [7]. Some other detection algorithms are based on color

978-1-4244-5040-4/10/$26.00 ©2010 IEEE

segmentation or geometrical features [8] [9], and finally on Distance Transform (DT) matching [10].

For traffic sign recognition, several machine-learning methods have been experimented, including Support Vector Machines [7], [8], a Bayesian generative model [6] and an Error-Correcting Output Code (ECOC) framework [5].

Most relevant to HISM is the work by Andriluka *et al.* [11]. There, the authors propose a part-based model for pedestrian detection in which each part votes for the object center. With our method, we overcome the need of manual subpart annotation: subparts are individuated automatically by a consistency segmentation rule. Moreover, we do not employ any supervised learning technique: our method is able to generalize recognition of the same sign with signs that consist of similar appearance and similar geometrical subparts distributions in an unsupervised manner. Furthermore, [11] learns a Gaussian distribution of the position of each manually labeled part relative to the object center that acts as a soft skeletal model; we do not impose any subpart spatial distribution and each subpart can independently vote for an object center.

III. Learning Human Signs

Our algorithm takes as input a picture of a sign, called *target sign*, that describes a desired destination. The method is then able to analyze any newly recorded image for the presence and the position of such a sign, probabilistically and without any hand-tuned models. Techniques like chamfer matching [12] or cross-correlation [13] aim at matching the exact visual appearance of the target sign with a test image. However, in cluttered and crowded environments these methods are prone to fail due to the weak description of the object. Moreover, the user might show a target sign not precisely aligned or totally visible. This can produce problems to a simple gradient matching method. Instead, we can achieve a higher level of robustness by describing a target sign with standard local image descriptors [14]. By matching such local descriptors, we obtain a far more reliable correspondence. Unfortunately, such an approach does not take into account that some parts of a sign are more important than others. As an example, numbers contain just a few features but they are a very descriptive part of a sign. Therefore, we developed a novel unsupervised matching method that overcomes these problems by producing a hierarchy in the object: a sign is automatically divided into subparts, described by standard local image descriptors; subparts compose a sign by defining geometrical constraints. Furthermore, the subpart description of our method allows importance weighting of each single subpart.

A. Features Extraction

The first step to robustly describe signs is to extract a set of local image descriptors. As we expect to match basic geometrical shapes contained in the signs, we make use of Shape Context descriptors [15] computed at Hessian-Laplace interest points [16]. This allows a quite dense description of the sign and, at the same time, a robust representation of local image regions. We denote interest points as vectors $\mathbf{x}_i = (x, y)$, their scales γ_i, and a set of interest points as $\mathcal{X}$. A descriptor computed at $\mathbf{x}_i$ is represented as a d-dimensional vector $\mathbf{h}_i$ that is part of a descriptor set $\mathcal{H}$.

It is important to notice that rotational invariance of the descriptors is not desired in our case, since we want to distinguish a 6 from a 9. We also note that our method is not only restricted to Shape Context descriptors, but can work properly with any other robust image features.

B. Sign Decomposition

In the next step of our algorithm, we divide a given sign into smaller subparts, called *tokens*. Signs are intrinsically designed to be clearly distinguishable by human eyes in any environment. For example, for traffic signs, the color schemes are regulated by an international convention [17]. Signs for public places generally follow the same rules. The colors used for the shapes are chosen to have a high contrast with respect to the background. By following these reasons, we define that uniform coherent and high-contrast areas delineate subparts. We thus proceed by binarizing the sign. Otsu's segmentation method [18] is able to accomplish this task in a robust and fast manner. It automatically selects the best thresholding value by minimizing the intra-class variance of foreground and background estimated pixels.

Then, a region growing algorithm [19] is applied to the active pixels of the binary image in order to group adjacent neighboring pixels into coherent regions. In order to remove inherent noise, we fix a minimum number of pixels σ per region. Finally, an agglomerative hierarchical clustering algorithm with average link [20] groups nearby regions into clusters by considering the distance between centers of gravity and using a threshold θ_d. This process avoids that a sign is decomposed into too many small regions. We then compute the bounding box B for each cluster and define the set of all interest points and corresponding descriptors inside B as a *sign token* $\mathcal{T}$:

$$\begin{aligned}
&\mathcal{T} := \{(\mathbf{x}_1, \mathbf{h}_1), \ldots, (\mathbf{x}_m, \mathbf{h}_m), B\} \\
&\text{where} \qquad \mathbf{x}_i \in B \quad \forall i = 1, \ldots, m \\
&\text{and} \qquad \mathbf{h}_i \text{ is computed at } \mathbf{x}_i \quad \forall i
\end{aligned} \tag{1}$$

It is important to remark that this method does not exploit the color or the shape of the cluster as a subpart descriptor, but it just uses the cluster as a coherent container for robust local image descriptors. Furthermore, our algorithm is not restricted to this procedure. Actually, any fast and robust segmentation method could be used at this stage.

C. Learning the Sign Hierarchy

Once a sign has been divided into tokens $\mathcal{T}$, we extract *geometrical information* in addition to the *appearance information* given by the descriptors $\mathbf{h}_i$ in each $\mathcal{T}$. An elegant and well established way to achieve this is by means of Implicit Shape Models (ISM) [1]. An Implicit Shape Model describes an object by a *codebook* of local appearance, i.e., a collection of local image descriptors, and the displacements between their associated interest points and the object center. In our case, this means that a codebook $\mathcal{C}^{\mathcal{T}}$ for a token $\mathcal{T}$ consists of all image descriptors $\mathbf{h}_i \in \mathcal{T}$ and all corresponding displacement vectors $\mathbf{v}_i$, where $\mathbf{v}_i = \mathbf{x}_i - \mathbf{c}^{\mathcal{T}}$ and $\mathbf{c}^{\mathcal{T}}$ is the center of gravity of $\mathcal{T}$. Thus, we define:

$$\mathcal{C}^{\mathcal{T}} := \{(\mathbf{h}_1, \mathbf{v}_1), \dots, (\mathbf{h}_m, \mathbf{v}_m)\}. \tag{2}$$

Now, using $\mathcal{C}^{\mathcal{T}}$, we can describe sign tokens, but for a reasoning on the higher level of signs, we need more information about the arrangement of the tokens in a sign. We proceed by introducing a hierarchical ISM of tokens, called *HISM*. A codebook $\mathcal{C}^{\mathcal{S}}$ of this hierarchical ISM is defined as the set of all tokens $\mathcal{T}_i$ of a sign $\mathcal{S}$ along with the displacements between the tokens' centroids $\mathbf{c}^{\mathcal{T}_i}$ and the center $\mathbf{c}^{\mathcal{S}}$ of the sign, and a weighting factor g_i, i.e.:

$$\begin{aligned} &\mathcal{C}^{\mathcal{S}} := \{(\mathcal{T}_1, \mathbf{w}_1, g_1), \dots, (\mathcal{T}_n, \mathbf{w}_n, g_n)\} \\ &\text{where} \qquad \mathbf{w}_i = \mathbf{c}^{\mathcal{T}_i} - \mathbf{c}^{\mathcal{S}} \ \forall i = 1, \dots, n \\ &\text{and} \quad g_i \quad \text{is the weight of} \quad \mathcal{T}_i \quad \text{(defined below)} \end{aligned} \tag{3}$$

At this point, we note that the boundaries of a sign $\mathcal{S}$ must be given to be able to compute its center $\mathbf{c}^{\mathcal{S}}$. Also, information about which tokens belong to which sign must be available to be able to create the codebook $\mathcal{C}^{\mathcal{S}}$. In this paper, we assume that a sign is always given in form of its bounding box. This ensures that $\mathbf{c}^{\mathcal{S}}$ and the tokens that belong to a sign are uniquely defined. This is nevertheless possible to remove this limitation with a clever segmentation algorithm, but this was not the focus of our work. The entire target sign processing is illustrated in Fig. 1.

D. Learning Token Weights

Using the HISM as described so far, we can compute a high-level codebook $\mathcal{C}^{\mathcal{S}}$ for each target sign and use it to find matching signs in new images presented to the robot. The details of this detection step are described in the next section. Before, however, we note that signs often differ only by a very small fraction, such as, e.g., one digit in the two signs for `Check-in 1` and `Check-in 3`. Using the HISM to match two signs like these would result in a high matching score, although the signs are significantly different. To address this issue, we additionally extract information about which tokens in a sign are most distinctive.

The intuition we use here is the fact that distinctive tokens occur only rarely in a given set of target signs. Assuming we are given a set of target signs $\mathfrak{S} = \{\mathcal{S}_1, \dots, \mathcal{S}_{m'}\}$. Now, for each sign $\mathcal{S}_i$ in $\mathfrak{S}$, we match all tokens $\mathcal{T}_j^i$ of $\mathcal{S}_i$ with all tokens from the other signs in $\mathfrak{S}$. As a result, we obtain a *matching score* s, defined below, for the correspondence between the tokens $\mathcal{T}_j^i$ and $\mathcal{T}_{j'}^{i'}$. This score is high if the tokens match well and low otherwise. A measure g_j of how well a token $\mathcal{T}_j^i$ matches *in general* can then be determined by summing up all matching scores:

$$g_j(\mathcal{T}_j^i) := \sum_{i'=1}^{m'} \sum_{j'=1}^{m(i')} s(\mathcal{T}_j^i, \mathcal{T}_{j'}^{i'}), \tag{4}$$

where $m(i')$ is the number of tokens encountered in sign $\mathcal{S}_{i'}$. Using this definition, the value of g_j for each token $\mathcal{T}_j^i$ characterizes its discriminative factor and is used at a later stage as a voting weight.

IV. Sign Detection

Whenever a user shows a target sign $\mathcal{S}$ to the robot, its task is to navigate to the goal position where $\mathcal{S}$ is encountered. To achieve this, it needs to match all new images with $\mathcal{S}$ and extract a direction indicator, i.e., an arrow, to infer its direction of motion. As a first step, the sign $\mathcal{S}$ is matched within the existing database $\mathfrak{S}$. In case $\mathcal{S} \notin \mathfrak{S}$, its HISM is computed and stored in $\mathfrak{S}$. Moreover, all the token weights g_j are updated.

Then, for a given new image I presented to the robot, all the interest points $\mathbf{x}_i^I$ and shape descriptors $\mathbf{h}_i^I$ are computed. These are then matched with all descriptors $\mathbf{h}_j$ found in the codebook $\mathcal{C}^{\mathcal{T}_k}$ of $\mathcal{S}$. The matching is carried out using a nearest neighbor distance ratio strategy [14]. The Euclidean distance $d(\mathbf{h}_i^I, \mathbf{h}_j)$ defines the matching score between $\mathbf{h}_i^I$ and $\mathbf{h}_j$. Let the first and second best matching descriptors be $\mathbf{h}_{j_1}$ and $\mathbf{h}_{j_2}$. A matching pair $(\mathbf{h}_{i_1}^I, \mathbf{h}_{j_1})$ is detected, if

$$d(\mathbf{h}_{i_1}^I, \mathbf{h}_{j_1}) \leq \vartheta_m \, d(\mathbf{h}_{i_1}^I, \mathbf{h}_{j_2}), \tag{5}$$

where ϑ_m is a distance ratio.

Each descriptor $\mathbf{h}_i^I$ that matches a descriptor $\mathbf{h}_j$ in a codebook $\mathcal{C}^{\mathcal{T}_k}$ casts a *vote* for an occurrence of the token $\mathcal{T}_k$ at the position

$$\begin{aligned} \mathbf{p}_{ij} &= \mathbf{x}_i - \mathbf{v}_j \, \delta_{ij} \\ \delta_{ij} &= \frac{\gamma_i}{\gamma_j}, \end{aligned} \tag{6}$$

where $\mathbf{x}_i$ and γ_i are the interest point and scale related to $\mathbf{h}_i^I$, $\mathbf{v}_j$ and γ_j are the stored displacement vector and scale related to $\mathbf{h}_j$, and δ_{ij} is the scale of the vote. Moreover, each vote is weighted inversely proportionally to the matching distance:

$$w_{ij} := \frac{1}{1 + d(\mathbf{h}_i^I, \mathbf{h}_j)} \tag{7}$$

All votes $\mathbf{q}_{ij} = (\mathbf{p}_{ij}, \delta_{ij})$ are then collected in a voting space $\mathcal{W}$. Occurrences of the token $\mathcal{T}_k$ are determined by finding high density loci in $\mathcal{W}$. To this end, we use mean shift mode estimation [21] with a spherical uniform kernel and a scale-adaptive bandwidth. This method starts at a random point $\mathbf{q} \in \mathcal{W}$ and iterates over computing the mean $\bar{\mathbf{q}}$ in a local vicinity of $\mathbf{q}$ and assigning $\bar{\mathbf{q}}$ to $\mathbf{q}$ until a minimal distance between $\mathbf{q}$ and $\bar{\mathbf{q}}$ is reached. The resulting $\bar{\mathbf{q}}$ is a *mode* $\mathbf{m}$ of the underlying points distribution and we call $\mathcal{M}$ the set of points $\mathbf{q}$ which support $\mathbf{m}$. The process is repeated until each point $\mathbf{q}_{ij}$ has been assigned to a mode $\mathbf{m}_{i'} = (\mathbf{p}_{i'}, \delta_{i'})$. The resulting modes yield *hypotheses* for the location of the token $\mathcal{T}_k$. The mode $\mathbf{m}_{\mathbf{i}'}$ defines the token $\mathcal{T}_{i'}$ with a matching score:

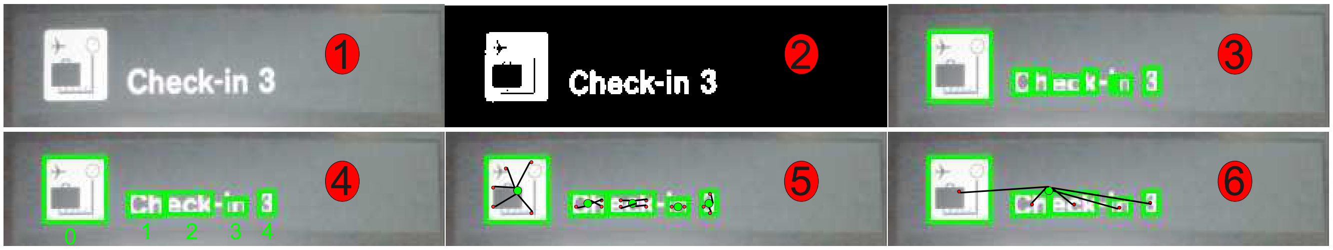

Fig. 1. Template processing. The input image 1 is binarized into image 2 with Otsu's segmentation method. A region growing algorithm finds clusters of coherent colors in image 2 and outputs them in image 3. An agglomerative clustering algorithm groups the regions of image 3 into tokens in image 4. Image 5 shows the codebooks for each token of image 4. Image 6 shows the codebook for the entire sign from the tokens codebooks of image 5.

$$s(\mathcal{T}_{i'},\mathcal{T}_k) := \sum_{\forall i,j\ q_{ij}\in\mathcal{M}_{i'}} w_{ij} \tag{8}$$

Finding the entire sign in I is then done in a similar fashion as for the tokens. A token $\mathcal{T}^I_{i'}$ that matches a token $\mathcal{T}_k$ of $\mathcal{C}^{\mathcal{S}}$ casts a vote for an occurence of the sign S at the position:

$$\begin{aligned} \mathbf{r}_{i'k} &= \quad \mathbf{p}_{i'} - \mathbf{w}_k\ \delta_{i'} \\ \zeta_{i'k} &= \qquad\quad \delta_{i'} \end{aligned} \tag{9}$$

where $\mathbf{p}_{i'}$ and $\delta_{i'}$ represent the hypothesis $\mathbf{m_{i'}}$ for the position and scale of the token $\mathcal{T}^I_{i'}$, and $\mathbf{w}_k$ is the distance vector stored in the codebook. Furthermore, each vote has a weight defined by:

$$v_{i'k} := s(\mathcal{T}^I_{i'},\mathcal{T}_k)\ \frac{1}{g_k} \tag{10}$$

As before, we collect the votes $\mathbf{t}_{i'k} = (\mathbf{r}_{i'k},\zeta_{i'k})$ that are hypotheses for possible locations of a sign $\mathcal{S}$ in a voting space $\mathcal{V}$. Again, we run mean shift mode estimation to find local density maxima in $\mathcal{V}$. The resulting location hypotheses $\mathbf{m}^{\mathcal{S}}_{j'} = (\mathbf{r}_{j'},\zeta_{j'})$ supported by $\mathcal{M}^{S}_{j'}$ for an occurrence of the sign $\mathcal{S}$ in I have a score defined by:

$$s(\mathbf{m}^{\mathcal{S}}_{j'}) := \sum_{\forall i',k\ \mathbf{t}_{i'k}\in\mathcal{M}^{S}_{j'}} v_{i'k} \tag{11}$$

The final score $s(\mathbf{m}^{\mathcal{S}}_{j'})$ can be compared to a detection threshold ϑ_d in order to validate the presence of the sign S in I at location $\mathbf{r}_{j'}$ and scale $\zeta_{j'}$. This threshold influences directly the performance of the sign detector: high values of ϑ_d reduce the number of false positive detections and thus increase the precision of the detector. Low values however increase the recall value. A detailed analysis on this is given in Sec. VI. We finally note that our system could be used in a probabilistic framework instead of outputting a binary answer.

V. Mapping Actions to Detections

We described in the previous sections how HISM represents a reliable sign matching method. In this section, we go a step further and analyze how it is possible to map actions to sign detections.

For a robot to navigate in a *sign-rich* environment like an airport, we not only need a reliable sign matching algorithm. We also have to associate the instruction related to the sign occurrence to a specific action of the robot, i.e., understand the direction arrows. We present two methods: in one we build an arrow detector, in the other we show how to exploit the full potential of HISM and infer it from the data.

A. Geometric Approach

A direct way of mapping actions to detections is to explicitly build an arrow detector. A simple arrow detector is described by an heuristic built on a given geometrical model, composed by piecewise linear segments. In order to detect an arrow, the image is segmented with Otsu's binarization algorithm. Regions with a certain minimal size and uniform color are extracted by using the same connected component approach as for *token* extraction. The following geometrical properties are taken into account for arrow detection:

- Aspect ratio of the bounding box computed on each extracted region.
- Filled ratio, i.e. number of pixels in the clustered region over the number of pixels of the bounding box.
- Horizontal or vertical symmetry axis of the cluster.
- Position of bounding box centroid.

We can efficiently compute these key geometrical properties by using the integral image technique introduced by [4]. The idea is to halve the bounding box by an horizontal line, and then by a vertical line. The integral image allows to easily compute the sum of the pixels in the subdivided bounding box areas, thus the symmetry ratios. We encode 8 types of arrows (up, down, left, right, and their 45-degrees rotated counterparts). We first detect the four cardinal arrows. Then we apply the same technique by rotating of 45-degree the detector for the remaining diagonal arrows. Classification is obtained by empirical thresholding on each of the geometrical property. Henceforth, this method introduces several free parameters to tune, lacks of generality, and is not adaptive.

B. Learning Approach

By using HISM, we can easily develop a robust and elegant method for mapping a sign to a robot action. The resulting algorithm is able to deal not only with arrows, but with any kind of geometrical shapes representing a direction, like triangles or complex direction symbols.

The idea is to collect with a robot several pictures of signs, and each time an image is taken, a label is associated. The label represents the high-level action that an operator has commanded to the robot: ”turn left”, ”turn right”, ”go up”, and so on. As soon as this dataset is built, all the images associated with the same action are collected into $\mathcal{I} = \left(\mathcal{I}^{up}, \mathcal{I}^{left}, \mathcal{I}^{right}, \mathcal{I}^{down}\right)$.

The intuition is to find the most recurrent *token* by simply comparing the signs of an image action set together. Thus, we consider each set $\mathcal{I}^i$ separately and detect all the signs in each set by running our algorithm on that. Therefore, similarly to the procedure of Section III, we produce a frequency analysis. All the *tokens* related to detected signs are discarded, therefore we count how many times each remaining *token* is repeated in $\mathcal{I}^i$ by computing the usual *token* matching score. In order to avoid ambiguities (e.g. another arrow of another sign in the image) we produce the assumption that the repeated tokens must be in an area close to a detected sign. We then store the *token* as a part of a sign, described by its ISM, related to the direction change $\mathcal{I}^i$. This procedure is run in for all $\mathcal{I}$.

VI. Experiments

We show in this section the results of our implementation and particularly analyze the qualitative and quantitative performance of our algorithms. Although we do not present an experiment involving a robotic platform, we outline in the conclusion the necessary steps for an integration.

All the experiments are based on a database of images collected at Zurich Airport with a standard digital camera. The original format of the images (JPEG, 3264x2448 pixels) is shrunk to a more reasonable 816x612 pixels BMP format. A gamma correction of 2 is also applied to increase the contrast. The majority of the images contain various signs at different scales, including background scenes of the airport. Some images contain no signs.

As mentioned before, our algorithm relies on several parameters: the descriptors matching distance ratio ϑ_m, the clustering threshold in the template image θ_d, the minimum area of a region in the template image σ, and the sign matching threshold ϑ_d. In the following experiments, we empirically fixed ϑ_m to 0.5, and made θ_d and σ scale dependent. The discriminative threshold ϑ_d was then iterated in order to find the optimal parameter.

A. Classification

In the first set of experiments we evaluate the general quantitative performance of our classifier in the case of target signs and arrows classification in random sample images.

1) Signs: As exposed above, the target signs are firstly matched against each other in order to determine the most distinctive token. In this experiment, we want to match the `Check-in 3` sign and we have 6 different target signs (`Check-in 1/2/3`, `Arrival 1/2`, `Railway`). The token containing the number `3` will get the higher voting weight after the matching process.

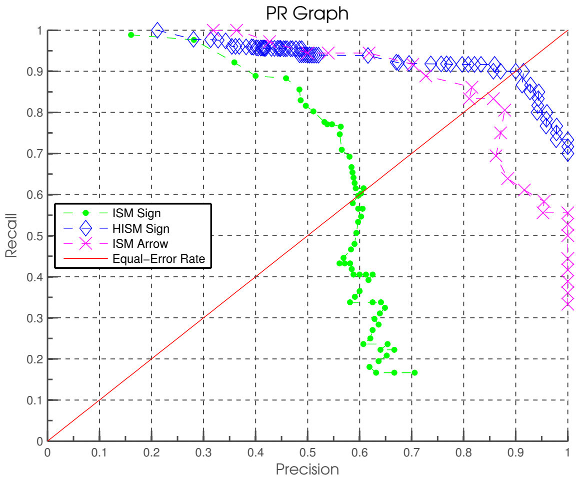

Fig. 2. Precision and Recall graph for signs and arrows classification. The plot shows that HISM outperforms ISM for signs detection, and that the arrows detection performs reliably.

The `Check-in 3` sign is matched with all the images of the database which is manually labeled to have a ground truth. The database contains 57 images with a `Check-in 3` occurrence out of 124 images.

The performance of the classification is evaluated with the Precision and Recall (PR) graph in Fig. 2, which shows the iterative discriminative threshold ϑ_d variation. The detection reaches 90 % at the Equal Error Rate (EER). For comparison purpose, Fig. 2 also reports the results using the standard ISM approach, which gets 60.25 % at EER. In this case, a lot of false positives are introduced, since `Check-in 1` and `Check-in 2` signs are counted as positives. Although we do not report their PR graph, we obtained similar results with the other target signs.

As a consequence of these experiments, we can state that our algorithm is able to correctly label the images belonging to the same path. Moreover, the accuracy of the detection might come to 100 %, if we refine our system by fusing multiple hypothesis of the same spot at different scales.

2) Arrows: In this second experiment, we aim at extracting the high-level navigation instructions contained in the signs. In our example of Zurich Airport, these instructions are represented by 8 types of arrows (right, left, up, down, and their 45-degrees rotated counterparts).

23 images containing an arrow pointing to the right are collected. They are matched against each other in order to extract their common token. The descriptors selected in each image are then again matched against each other. The token which gets the highest summed matching score will represent this kind of arrow. The extracted token is finally matched with the entire image collection to assess the performance of the arrow classification. The database contains 35 occurrences of right arrows out of 124 images.

The results are expressed in the PR graph in Fig. 2.

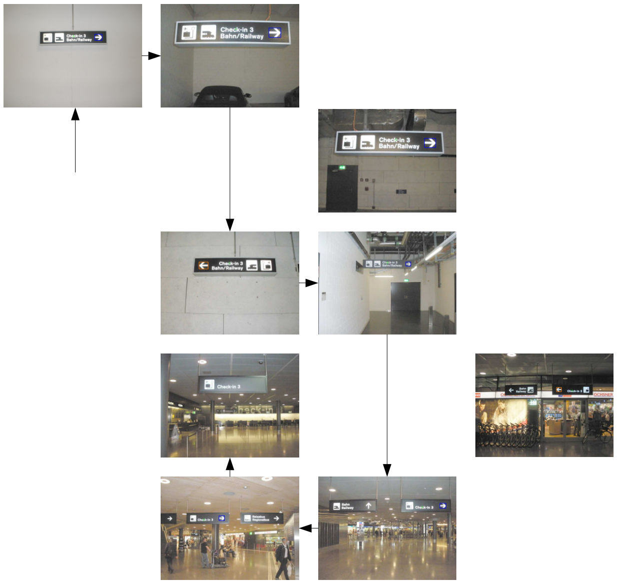

Fig. 3. Example of exploration and mapping in Zurich Airport. The robot starts from the garage and is able to reach the *Check-In 3* by following the instructions on the sign. In the end, a topological map is also built.

The detection reaches 83.33 % at EER. We believe that these results could also be improved by using temporal integration. Although we only show here the classification of right arrows, these results are generalizable to any kind of direction instructions.

For comparison purpose, we also assess the performance of our geometrical arrow detector on the same set of images. We obtain an *accuracy* of 93.54 %, which is slightly better than with the learning method. However, this approach is computationally more expensive and not as easily scalable in other environments.

B. Exploration and Topological Map Creation

In this experiment, we show how the combination of the previous results can yield an interesting robotic application. Assuming that a robot has an arrow and a sign detector, we can for instance put it in the garage of the airport in front of a direction sign and show him an image of the `Check-in 3` destination sign it has to reach. The robot uses the sign detector to find which part in the current image is relevant to the destination. It then maps the direction instruction to a motor action of its base. The result of this action leads him to the next sign until it reaches the destination. Fig. 3 qualitatively shows the result of this experiment.

This experiment shows that our algorithm is not only suitable for finding its way in an airport. It can also be used with a robot to build a topological map of the airport.

VII. Conclusion

In this paper, we introduced the concept of Hierarchical Implicit Shape Models for robustly recognizing and generalizing over different kinds of signs in unstructured environments. It consists of a two-level hierarchy: one based on image primitives and the other on subparts of signs with weights learned from data. We showed that with HISM, it is easy to map semantics to detections by learning the meaning of the direction arrows in an unsupervised manner. Experiments have been conducted from datasets retrieved in a crowded airport terminal, that show typical guiding robot applications, with promising results.

As future work, we aim to extend this approach by using temporal integration and to develop learning and detection into a robotic mobile platform ready to be deployed in man-made environments.

Acknowledgment

This work was funded within the EU Projects BACS-FP6-IST-027140 and EUROPA-FP7-231888.

References

[1] B. Leibe, A. Leonardis, and B. Schiele, “Robust object detection with interleaved categorization and segmentation,” *Int. J. Comput. Vis.*, vol. 77, no. 1-3, pp. 259–289, 2008.

[2] G. Qingji, Y. Yue, and Y. Guoqing, “Detection of public information sign in airport terminal based on multi-scales spatio-temporal vis. information,” in *Proc. Int. Conf. Comput. Sci. Softw. Eng.*, 2008.

[3] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.

[4] P. Viola and M. Jones, “Robust real-time face detection,” *Int. J. Comput. Vis.*, vol. 57, no. 2, pp. 137–154, 2004.

[5] X. Baro, S. Escalera, J. Vitria, O. Pujol, and P. Radeva, “Traffic sign recognition using evolutionary adaboost detection and forest-ecoc classification,” *IEEE Trans. Intell. Transport. Syst.*, vol. 10, no. 1, pp. 113–126, 2009.

[6] C. Bahlmann, Y. Zhu, V. Ramesh, M. Pellkofert, and T. Koehled, “A system for traffic sign detection, tracking, and recognition using color, shape, and motion information,” in *Proc. IEEE Intell. Veh. Sym.*, 2005.

[7] R. Timofte, K. Zimmermann, and L. Van Gool, “Multi-view traffic sign detection, recognition, and 3d localisation,” in *Proc. IEEE Wksh. App. Comput. Vis.*, 2009.

[8] C. Kiran, L. Prabhu, R. Abdu, and K. Rajeev, “Traffic sign detection and pattern recognition using support vector machine,” in *Proc. Int. Conf. Adv. Pattern Recog.*, 2009.

[9] H. Huang, C. Chen, Y. Jia, and S. Tang, “Automatic detection and recognition of circular road sign,” in *Proc. IEEE/ASME Int. Conf. Mechatron. Embedded Sys. App.*, 2008.

[10] D. Gavrila and V. Philomin, “Real-time object detection for smart vehicles,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 1999.

[11] M. Andriluka, S. Roth, and B. Schiele, “People-tracking-by-detection and people-detection-by-tracking,” in *Proc. of the IEEE Conf. on Comput. Vis. and Pattern Recog.*, 2008.

[12] H. Barrow, J. Tenenbaum, R. Bolles, and H. Wolf, “Parametric correspondence and chamfer matching: Two new techniques for image matching,” in *Proc. Int. Joint Conf. Art. Intell.*, 1977.

[13] R. B. Fisher and P. Oliver, “Multi-variate cross-correlation and image matching,” in *Proc. British Machine Vis. Conf.*, 1995.

[14] K. Mikolajczyk and C. Schmid, “A performance evaluation of local descriptors,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 27, no. 10, pp. 1615–1630, 2005.

[15] S. Belongie, J. Malik, and J. Puzicha, “Shape matching and object recognition using shape contexts,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 24, no. 24, pp. 509–522, 2002.

[16] K. Mikolajczyk and C. Schmid, “Scale and affine invariant interest point detectors,” *Int. J. Comput. Vis.*, vol. 60, no. 1, pp. 63–86, 2004.

[17] Economic Commission for Europe, Inland Transport Committee, “Convention on road signs and signals,” 1968.

[18] N. Otsu, “A threshold selection method from gray-level histograms,” *IEEE Trans. Syst., Man, Cybern.*, vol. 9, no. 1, pp. 62–66, 1979.

[19] S. Hojjatoleslami and J. Kittler, “Region growing: A new approach,” *IEEE Trans. Image Processing*, vol. 7, no. 7, pp. 1079–1084, 1998.

[20] P. Berkhin, “A survey of clustering data mining techniques,” in *Grouping Multidimensional Data*. Springer, 2006, pp. 25–71.

[21] D. Comaniciu and P. Meer, “Mean shift: A robust approach toward feature space analysis,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 24, no. 5, pp. 603–619, 2002.