Detection and Classi£cation of Gateways for the Acquisition of Structured Robot Maps

Derik Schröter, Thomas Weber, Michael Beetz and Bernd Radig

Munich University of Technology
Boltzmannstr. 3, 85748 Garching b. München, Germany
schroetd@in.tum.de, webert@in.tum.de, beetz@in.tum.de
http://www9.in.tum.de/people/schroetd/Research/

Abstract. The automatic acquisition of structured object maps requires sophisticated perceptual mechanisms that enable the robot to recognize the objects that are to be stored in the robot map. This paper investigates a particular object recognition problem: the automatic detection and classi£cation of gateways in of£ce environments based on laser range data. We will propose, discuss, and empirically evaluate a sensor model for crossing gateways and different approaches to gateway classi£cation including simple maximum classi£ers and HMM-based classi£cation of observation sequences.

1 Introduction

So far robot maps primarily support safe and ef£cient navigation [2, 7], see [11] for an extended overview of state-of-the-art mapping approaches. The next generation of maps will in addition provide better support for the achievement of service tasks. They will do so by explicitly representing the environment structure and by modeling relevant objects of the environment.

In our previous research, we have proposed Region & Gateway Maps (RG Maps) as resources for autonomous mobile robots acting in structured human indoor environments [4]. RG maps are tuples $\langle R, G \rangle$, where R denotes a set of regions and G is a set of gateways that represent the possible transitions between regions. A region has a compact geometric description, a bounding box, a list of adjacent gateways, and a set of models that represent the task relevant objects within the region. The second key component of RG maps are gateways, prominent and recognizable areas that connect different parts of the robot's environment. The recognition of gateways allow robots to autonomously extract the environment structure and represent it in the map [8, 3, 1].

In two companion papers we have detailed our mechanisms for acquiring compact geometric descriptions of regions [4] and for the acquisition of models of rectangular task relevant objects [10]. This paper addresses the problem of automatically detecting and classifying crossing gateways.

Gateways form perceptually recognizable, characteristic transitions between two or more adjacent regions. They can be traversed in any direction and are the only possibility to pass from one region into another. The partitioning of ¤oor plans is based on gateways such as cross-ways, junctions, turns and narrow passages, see also £gure 1. In our approach gateways are speci£ed by a class label, adjacent regions, traversal directions, crossing-points and gateway-points that can be used for detecting when a gateway is entered and left. The set of discrete **gateway points** is derived from features extracted from a single laser scan (see section 2). Pairs of these gateway points form passages, the robot can pass through. Narrow passages or open-close-transitions are characterized by a single pair of gateway points, whereas multi-passage gateways like junctions for example contain several passages. It is also possible to combine multiple gateway structures as encountered in of£ce environments (refer to Fig. 1). We will focus here on crossing gateways, i.e. gateways which connect hallway regions. The detailed concepts and properties of such gateways can be found in [4].

The computational problem of gateway recognition and classi£cation can be formulated as follows: Given a single scan or a sequence of scans provided by a laser range £nder and a set of gateway models, the robot autonomously detects and classi£es crossing gateways. We will solve the gateway recognition problem in a computational process that executes a sequence of three steps: **(1)** Generating hypotheses for virtual line models (VLMs) (sec. 2), **(2)** Determining weights according to general and speci£c gateway models (sec. 3), **(3)** Using the generated observation vector for classi£cation (sec. 5). Finally, we empirically evaluate the proposed methods (sec. 5) and conclude.

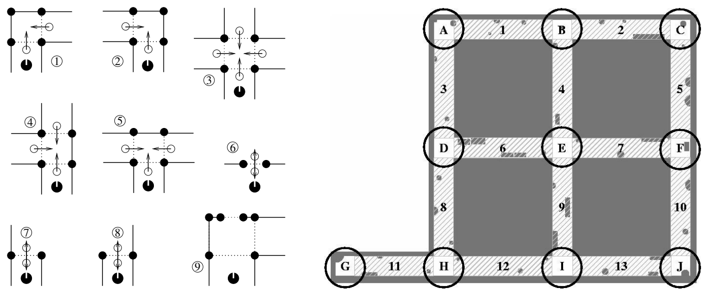


Fig. 1. :**Left**: Classes of Gateways - Right/Left Turn (1,2), X-Crossing (3), T-Junction/Forking (4,5), Narrow Passage (6), Right/Left Opening (7,8), Combination of Gateways (9); (• gateway point, ◦ crossing point, ← traversal direction, **...** region border); **Right**: Example environment - letters denote gateways, numbers denote regions

2 Generating Hypotheses for Virtual Line Models (VLMs)

In order to represent gateway hypotheses we propose virtual line models (VLMs) as an appropriate feature language. VLMs are based on the assumptions that environments are rectangular and hallways have approximately the same width. The VLM consists of a left, right and front virtual line as well as a hidden virtual line (Fig. 2). To generate VLM hypotheses, we £rst extract low-level features, i.e. virtual lines and depth singularities from the line segment and point scan, respectively. In the next step, those virtual lines are grouped to form hypotheses with respect to the VLM in Fig. 2. In the £rst processing step the algorithm generates a line segment scan L_{LS} from the point scan L_P by the means of linear regression according to [6].

Virtual Lines. Line segments from L_{LS} which lie approximately on the same line are grouped and represented by that line, also referred to as virtual line (see Fig. 2).

Depth Singularities. This point feature is extracted from L_P and denotes discontinuities in the distance measurements of a laser scan, see Fig. 2. The parameter Δd_{min} indicates the minimum distance difference of two succeeding distance measurements to represent a depth singularity. $P_i \in L_P$ is the point where the distance measurement d_i ends. P^i_{ds} are the points at the depth singularities.

$$P^i_{ds} = \{P_i \; : \; (d_i < d_{i\pm1}) \; \wedge \; (|d_i - d_{i\pm1}| > \Delta d_{min})\}$$

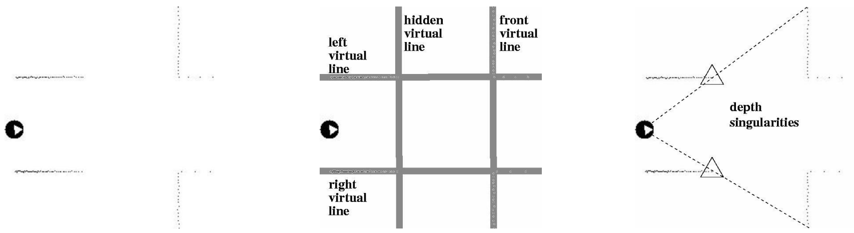


Fig. 2. Point scan of an X-Crossing (left); Virtual Line Model for Crossing Gateways, i.e. X-Crossing, L/R-Turn, T-Junction (middle); Depth singularities in a laser scan (right)

Virtual Line Grouping. Based on the virtual lines and depth singularities we generate hypotheses for VLMs, which signal that a crossing gateway of some kind may be present. To generate candidates for the virtual left and right line, we search for parallels among the virtual lines, where the robot is in between. Virtual front lines intersect a pair of parallels approximately in a right angle and in front of the robot. Finally, we estimate the virtual hidden lines. Therefore, we consider depth singularities, that are close to the virtual left or right line. The hidden line is constructed such that it is parallel to the virtual front line and intersects with the given depth singularity. To deal with situations where no valid depth singularities are present, we add hypotheses where the estimation of the hidden line is solely based on the environment assumptions. As a result we obtain a set of annotated virtual line quadruples, which represent hypotheses for VLMs. The gateway points are de£ned by the intersections of those virtual lines.

3 Evaluating the VLM Hypotheses

We evaluate gateway hypotheses by assessing the similarity of a perceived VLM and a speci£c gateway class. Therefore, we propose the following measures:
1. rectangularity and distance measure to re¤ect the general model quality and
2. freespace measure to account for the match with a speci£c gateway class.
As a result we obtain an observation vector for each VLM hypothesis. Additionally, we track VLM hypotheses over consecutive measurements while the robot is moving towards the gateway to generate observation sequences.

Distance Measure. The expected hallway width d_{hw} has been manually measured. Deviations from this value are weighted according to:

$$w^i_{distance} = 1 - \sqrt{\frac{|d_i - d_{hw}|}{d_{hw}}} \qquad W_d = \frac{\sum_{i=1}^{4} w^i_{distance}}{4}$$

Whereas d_i is the Euclidian distance between two neighboring gateway points and W_d denotes the averaged distance weight.

Rectangularity Measure. The rectangularity criterion refers to the inner angles α_i ($i = 1...4$) of the convex quadrangle, given by the VLM. We de£ne the rectangularity by the deviation of the inner angles α_i from $\frac{\pi}{2}$.

$$W_r = 1 - \frac{\sum_{i=1}^{4} |\alpha_i - \frac{\pi}{2}|}{2\pi}$$

Freespace Measure. Considering the VLM as depicted in Fig. 2 (right), we de£ne three pairs of gateway points, namely on the virtual left, right and front line. According to those pairs of gateway points, we divide the sensor data into three sectors, Fig. 3.

Each sector S_i comprises N_{S_i} measurements. Based on those de£nitions we propose the freespace measure (FSM) as a quantity for the match of a hypothesis to the sensor data. In each of the three sectors the sensor measurements should either be close to a given line (*On Line FSM*) or should cross a given line (*Over Line FSM*). The gateway class determines which of the two FSM variants applies to a certain sector. For example, considering an *L-Turn* the measurements in the front sector are expected to match the virtual front line (*On Line FSM*). Whereas for an *X-Crossing*, measurements in the same sector are expected to cross the virtual front line (*Over Line FSM*), see Fig. 3.

On Line FSM. P_i denotes a laser measurement from the point scan L_P and $d(P_i)$ is the respective distance measurement. We compute a point P_i^{vl} on the considered virtual line and its distance to the robot $d(P_i^{vl})$, whereas P_i and P_i^{vl} lie on the same ray from the robot. Then we count all measurements for which the difference of $d(P_i)$ and $d(P_i^{vl})$ lies between a given lower and upper threshold. Finally, we normalize this online-count (C_{ol}) with the overall number of measurements in the sector:

$$W_{on\ line}^{S_i} = C_{ol}^{S_i} / N_{S_i}$$

Over Line FSM. This measure only applies to the front sector. We construct a line l_{par} parallel to the virtual front line and set back by a given distance. Then we count all measurements which intersect l_{par}, and normalize the resulting over-line-count. Analogous to the *On Line FSM* we get $W_{over\ line}^{S_i}$, see also Fig. 3.

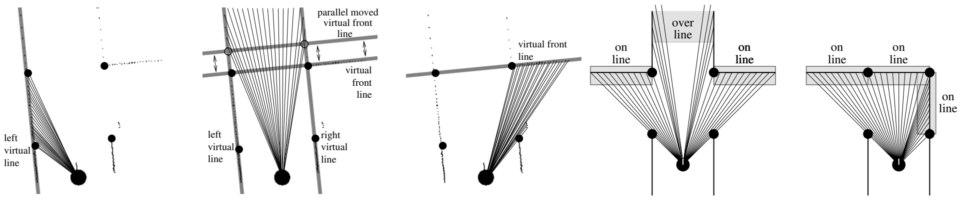


Fig. 3. From Left to Right: Freespace measure (FSM) for different cases - *On Line FSM* (1,3), *Over Line FSM* (2), (● gateway point, — laser scan measurement, - - line for free space evaluation); FSM con£gurations for *XCrossing* (4) and *LTurn* (5)

Generating Gateway Weights and Observation Sequences. Utilizing the proposed measurements, we de£ne weights for each VLM hypothesis with regard to a certain gateway class GW:

$$W(GW, VLM) = \frac{f_{vlm}}{2} \cdot (W_d(VLM) + W_r(VLM)) + \frac{f_{gw}}{3} \cdot \sum_{i=1}^{3} W_{FSM}^{S_i}(GW)$$

Whereas f_{vlm} and f_{gw} denote weighting factors for the general and gateway speci£c measurements, respectively. As a result we obtain an observation vector for each VLM hypothesis, where the entries quantify the similarity of the hypothesis to a speci£c gateway class. In most practical cases the mobile platform approaches the gateway area. Thus, we observe the same VLM hypotheses from different positions, where the distance to the gateway is continuously decreasing. The VLM hypotheses tracking is based on the gateway points and Euclidian distances. If all gateway points of two VLM hypotheses have an approximate match, they are considered to be identical. Based on this tracking, we obtain sequences of observation vectors. A sequence starts when the hypothesis is £rst observed and the distance falls below a threshold. It is £nished or corrupted when it is either lost or the robot enters the gateway.

4 Classi£cation

We now investigate the computational task of classifying the obtained observation sequences with regard to the introduced gateway classes, by means of the following classi£cation methods: based on the observation vector closest to the gateway, weighted average over all observation vectors in a sequence and Hidden Markov Models.

1. Single Observation and Averaged Sequence based Classi£cation

Observations close to a gateway imply a more complete coverage of the gateway area by the sensors, hence they are in general the most informative. The single observation classi£er (SOC) considers the maximum weight to determine the gateway at hand. This approach demonstrates the discrimination power of the freespace measurements and the resulting weights. It is, however, very sensitive to sensor noise, occlusions and dynamic changes in the environment. A simple alternative is the fusion of consecutive measurements by calculating a weighted average over the observation sequence, where the weights are inversely proportional to the distance. Afterwards, SOC is used to decide which speci£c gateway is present. Whereas the approach considers the complete observation sequence, it does not fully exploit probabilistic properties of observations and temporal relations between them.

2. Classi£cation based on Hidden Markov Models (HMMs)

A more promising approach to gateway classi£cation is the use of HMMs. They provide mechanisms to model temporal structures in sequences, by the use of probabilistic observation and state transition models. A detailed description of the theory can be found in [9]. In the next paragraphs we brie¤y outline the steps necessary to use HMMs in the context of gateway detection based on the introduced sensor model.

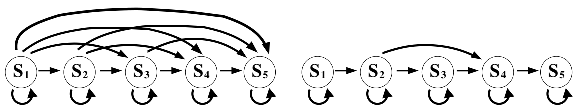
Clustering the Data and Initializing the HMM. Since our sensor model provides continuous measurements we use HMMs with continuous outputs. To deal with the implicated complexity of such HMMs we explicitly cluster the data using the k-means-algorithm [5]. The clusters are then used to built the observation model (mean and covariance matrices), and to de£ne the structure of the HMM. Since the coverage of the gateway area by the sensor differs for different positions, we expect the data to represent clusters for different distance intervals, and we compute the start values for the k-means-algorithm accordingly. This assumption is veri£ed by the fact that the mean values are only altered slightly by the k-means clustering. To get a further intuition we labeled each observation vector according to distance intervals: I_1, I_2, I_3. After the clustering we sorted the clusters according to the intervals, and counted how much of the prelabeled data has been assigned to which cluster. In Table 1 it can be seen that all of the clusters contain a reasonable amount of samples (over all sum), and that clusters are built according to distance intervals (max/min distance). They contain either observations from disjunctive or slightly overlapping distance intervals or represent different distributions for the same interval. Those £ndings are very important for the choice of the HMM structure and initialization, but they also allow for interpretation of the learned model.

To initialize the HMM all clusters C_k that cover approximately the same distance interval are assigned to the same HMM state S_i. More precisely, the covariance matrix and the mean of each C_k add a dimension to the observation model of S_i. Considering Table 1, we obtain an HMM with £ve states, where $[C_1, C_2, C_3]$ present the £rst state, C_4 the second, $[C_5, C_6]$ the third, C_7 and C_8 the fourth and £fth, respectively. The mixture matrix M_{mix} is initialized uniformly, the dimension is given by the number of

cluster id	1	2	3	4	5	6	7	8
$I_1 = 7m...4m$	475	508	685	795	2	1	0	0
$I_2 = 4m...2m$	0	0	0	16	277	252	1143	44
$I_3 = 2m...$	0	0	0	0	0	0	32	437
over all sum	475	508	685	811	279	253	1175	481
mean distance	6112	5796.4	5770.2	4455	3796.8	3538.3	2706.2	1755.5
max distance	6872.2	6956.1	6976.3	4968.7	4009.4	4103.6	3591.7	2310.3
min distance	4854.5	4960.8	4878	3954.8	3438.6	3037.8	1775.9	1427.9

Table 1. Clustering for T-Crossing data, columns refer to different cluster, rows depict cluster properties; all distance measurements in millimeter

states Q and the maximum number of mixture components M. The states are arranged to form a left-right HMM, and according to the sequences, left refers to large and right to small distances. Although, in a left-right HMM consequently all entries below the diagonal of the transition matrix T are zero, we initialized the full matrix with 1/Q. Fig. 4 (left) shows a left-right model, where the arrows denote the possible transitions from state S_i to S_j with probability p_{ij}.



$$\begin{pmatrix} 0.8 & 0.2 & 0 & 0 & 0 \\ 0 & 0.58 & 0.417 & 0.003 & 0 \\ 0 & 0 & 0.37 & 0.63 & 0 \\ 0 & 0 & 0 & 0.68 & 0.32 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Fig. 4. Graph of left-right HMM (left) and learned HMM for case presented in Table 1 (right); Also given is the transition matrix T, presenting the respective transition probabilities p_{ij}.

Learning and Evaluation of the Hidden Markov Model. We £x the observation model obtained from clustering and use expectation-maximization (EM) learning to determine appropriate values for T, M_{mix} and the state prior, according to [9]. Since the observation space given by our sensor model is £lled very sparsely and the covariances of the data are all considerably small, we encountered problems of over£tting. That means, observation probabilities tend to zero and cause numerical instabilities. To anticipate those problems, we add noise to the clustering data, to arti£cially spread the distributions. The task to £nd the best HMM is to optimize the learning with regard to the number of clusters and the noise to be added. Too many clusters cause some clusters to not cover a suf£cient number of samples, and too few reduce the discrimination power. On the other hand too much noise reduces the discrimination power but increases the generality of the model. None or little noise results in over-selective HMMs. By now we semiautomatically search for an optimal solution. Fig. 4 (right) shows the graph of the HMM and its transition matrix that were learned for the case presented in Table 1. As expected we obtained a left-right model (no backward transitions), and most states are only connected with the next state.

5 Experimental Results

In this section, we will empirically evaluate the proposed approaches. To acquire a suf£cient amount of data for different hallway environments we used a simulator which provides laser measurements, based on the sensor model of the real SICK LMS200 laser range £nder. Also, we annotated the maps, in order to automatically label the recorded observation sequences. As a result we obtained about 40000 observation vectors for six different environments, which adds up to approximately 6000 observation sequences. The environments differ in the amount of clutter that is present, and the width of hallways (2 meter and 3 meter). The environment depicted in Fig. 1 is referred to as *2m uncluttered*. For images of all environments refer to our homepage.

Gateway class	XCrossing	TCrossing	LTCrossing	RTCrossing	LTurn	RTurn
2m uncluttered	403	503	235	224	99	104
averaged seq	100%	88.8668%	99.1489%	100%	5.05051%	24.0385%
last obs	98.7593%	100%	93.1915%	97.3214%	50.5051%	100%
2m slightly cluttered	291	210	201	146	82	94
averaged seq	99.6564%	52.8571%	100%	100%	2.43902%	5.31915%
last obs	99.3127%	100%	96.5174%	100%	78.0488%	98.9362%
2m cluttered	148	106	79	60	26	34
averaged seq	95.2703%	53.7736%	100%	93.3333%	11.5385%	2.94118%
last obs	93.9189%	90.566%	83.5443%	93.3333%	46.1538%	94.1176%

Table 2. Classi£cation results for the single observation classi£er (last obs) and the averaged sequence classi£er (averaged seq); *DeadEnds* have 100% recognition rate for all cases.

It can be seen from Tab. 2, that in most cases the recognition rate for the single observation classi£er is very high. As expected the classi£cation is slightly worse, when clutter is present, due to ambiguous measurements. The situation is similar for the sequence average classi£er. It is important to note, that the weight from the hallway detector had to be ignored in several cases. We conclude, that for semi-cluttered ("friendly") environments, we can expect good recognition results, but the classi£er gives no evidence of how good the decision was.

The EM learning converged to a left-right model with expected apriori probabilities for all types of sequences and data from the uncluttered environments. Since we did not yet obtain an optimal set of HMMs to handle all gateway types, we examined the classi£cation rate for pairs of gateway classes and different environments. We found that the hallway width does not in¤uence the discrimination power, but as for the simple classi£ers, the recognition rate decreases in the presence of clutter.

When considering test data, that refers to that same environment, the classi£cation is correct for all types of sequences.

This shows, that HMMs can perform very well on the given data. When we appropriately adjust the two parameters described in sec. 4, the number of clusters and the amount of noise, we can increase both the discrimination power in the pairwise comparison and the tolerance against clutter. These results suggest, that an optimal solution for all HMMs may exist. That means, the next step is to de£ne optimization criteria, and £nd feasible algorithms to solve that problem for all HMMs at the same time. The main reason for us to investigate this approach is, that the resulting HMM based classi£er provides probabilities for observation sequences with regard to the different gateways. Thus, it is possible to globally fuse the results of different observation sequences in a very formal way by the means of a Bayes £lter.

6 Conclusion

In this paper we proposed a sensor model for the detection and classi£cation of different classes of crossing gateways. The model is based on the virtual line model (VLM) and different general and gateway speci£c measures, that enable us to assess the similarity of the perceived sensor data and the different gateway classes. As a result we obtained observation sequences for when the robot is approaching gateway areas. We investigated the properties of that data, and showed that it is a discriminating feature language well suited for the given task. We proposed three classi£ers, based on the generated observation sequences. The simple classi£ers perform well in slightly cluttered and static environments, but lack a quantity for how good the classi£cation was. On the other hand, we presented theory and experiments for the HMM based sequence classi£cation. And it could be seen, that the approach is very promising, with regard to global fusion of observations and reasoning under uncertainty. This is due to the property of HMMs to provide probabilities for any given sequences, and hence a notion of how sure the classi£er is about its decision.

The next step is to learn the set of HMMs for all classes of gateways, so as to maximize the discrimination power across the set of HMMs, and also the tolerance to changes in the environment.

References

1. P. Beeson, M. MacMahon, J. Modayil, J. Provost, F. Savelli, and B. Kuipers. Exploiting local perceptual models for topological map-building. *IJCAI-2003 Workshop RUR-03*.
2. W. Burgard, A.B. Cremers, D. Fox, D. Hähnel, G. Lakemeyer, D. Schulz, W. Steiner, and S. Thrun. Experiences with an interactive museum tour-guide robot. *Arti£cial Intelligence*, 114(1-2), 2000.
3. Eric Chown. Gateways: An approach to parsing spatial domains. In *ICML 2000 Workshop on Machine Learning of Spatial Knowledge*, 2000.
4. J.-S. Gutmann D. Schröter, M. Beetz. RG Mapping: Learning Compact and Structured 2D Line Maps of Indoor Environments. In *Proc. of 11th IEEE ROMAN 2002, Berlin/Germany*.
5. R. O. Duda, P. E. Hart, and D. G Stork. *Pattern Classi£cation*. New York: JohnWiley & Sons, Inc., second edition., 2001.
6. J.-S. Gutmann. Robuste Navigation autonomer mobiler Systeme (in German). Akademische Verlagsgesellschaft Aka, Berlin, 2000. Doctoral Thesis *University of Freiburg*.
7. D. Haehnel, D. Fox, W. Burgard, and S. Thrun. A highly ef£cient FastSLAM algorithm for generating cyclic maps of large-scale environments from raw laser range measurements. In *Proc. of IEEE IROS, Las Vegas/USA*, 2003.
8. David Kortenkamp. *Cognitive Maps for mobile robots: A representation for mapping and navigation*. PhD thesis, University of Michigan, 1993.
9. L. R. Rabiner and B. H. Juang. An introduction to hidden Markov models. *IEEE ASSP Magazine*, pages 4–15, January 1986.
10. D. Schröter and M. Beetz. Acquiring Modells of Rectangular Objects for Robot Maps. In *Proc. of IEEE ICRA, New Orleans/USA*, 2004.
11. S. Thrun. Robotic mapping: A survey. In G. Lakemeyer and B. Nebel, editors, *Exploring Arti£cial Intelligence in the New Millenium*. Morgan Kaufmann, 2002.