A Feature-Based Approach for Matching Laser Range Scans to Schematic Maps

Diedrich Wolter
Department for Informatics
University of Hamburg
D-22527 Hamburg, Germany
wolter@informatik.uni-hamburg.de

5.4.2001

Abstract

Navigating a mobile robot instructed with a map involves a variety of tasks. Besides route planning, it is especially challenging to align map and environment. Perceptual information about shape of the passable space need to be accumulated and transformed into a suitable representation that allows for both, data- and model-driven recognition processes. Successful shape matching techniques that originated from computer vision serve as a starting point in deriving a new shape similarity measure suitable to match shape information perceived from different viewpoints and to align perceived shapes and shapes represented in the schematic map. A meaningful decomposition allows to avoid building a global metric map and the difficulties involved. Though only the shape processing described has been tested in a real-world environment, results seem promising.
Keywords: map, navigation, shape, scan-matching

1 Introduction

Schematic maps are especially interesting for robot instruction in service robotics: they can be produced easily by a human instructor and are well-suited for communication [3]. The key point in performing map-based navigational tasks is to align map and environment and to match them against each other. However, handling these geometrically imprecise, simplified maps is a non-trivial task. Schematic maps provide shape information which can also be perceived from the environment (e.g. via laser range finder). We will investigate how shape-matching techniques that originated from computer vision can be applied to identify map-correspondences and to aid self-localization. Our intention is to connect the two fields: the processing of perceptual data and the use of an abstract schematic map. Shape representations allow to achieve a more general view on the data involved in this domain. Our realization of navigation does not differ much from other approaches. It is dealt with upon a meaningful decomposition proposed, avoiding the difficult accumulation of a global map.

2 Feature-Based Approach

Schematic maps provide an object oriented spatial representation, they are made of line strokes which form environmental features. Our starting point is a drawn map with direct access to the polygonal lines the map is made of. See figure 1 for an example of such a map. Annotations besides start and goal position are not taken into account as this is beyond our scope. To navigate a common robot equipped with a laser range finder and both, instructed and aided by a schematic map, there are two possibilities.

The most common approach is using standard scan matching algorithms to first build up a global metric map. Reduction of map data before path planning is considered necessary, as overlapping scans

Proc. TIMR 01 - Towards Intelligent Mobile Robots, Manchester 2001.
Technical Report Series, Department of Computer Science, Manchester University, ISSN 1361 - 6161.
Report number UMCS-01-4-1. http://www.cs.man.ac.uk/csonly/cstechrep/titles01.html

tend to blur the map [10]. The blurring occurs due to noisy data and properties of optimization techniques involved in scan matching algorithms. The constructed representation, an occupancy grid, serves as a basis for various navigation strategies. In many approaches to robot navigation these global metric maps serve as a starting point, even though the representation used for path planning is no longer metric. Among others, see for example [6, 10]. These approaches may then be extended to use maps by matching the pictorial representations of exploration and instruction map against each other. Therefore, a pixel based correspondence (e.g. Hausdorff distance) may be used [11]. However, there are several disadvantages in proceeding this way. Besides being confronted with all the difficulties in building a global metric map, which itself has not been solved satisfactorily yet, this approach ignores a schematic map's imprecision. It relies completely on matching schematic map and exploration map exactly. Therefore geometrically rather precise maps need to be provided, as pixel-based correspondences ignore the presence of non-local rather conceptual features like corners. So applicability for instructing robots is limited.

The second possibility in map based navigation is to interpret sensory data as features. Computing correspondences on a more abstract level of shape features allows to combine model- and data-driven recognition processes. Therefore, the instruction map's imprecision and incompleteness can be taken into account. We investigate the interpretation of schematic maps in terms of meaningful features. Standard approaches in map building from laser range finder data apply numerical optimization techniques to correlate data with the partial map built so far, see [5] for an overview. These maps are simply registrations of reflection points in the plane. Multiple readings of the same surrounding tend to blur the map, as approximation fails to account for local noise. In contrast to these approaches, we represent scan data as shape data, namely polygonal lines specifying the shape's contour. Computing correspondences between perceived information and the map is done by finding the most similar partial shape at a nearby position. A novel shape similarity measure has been designed for this task. Shape matching produces perfect matches, so no blurring occurs. Furthermore, perception maps made of shape information need not be reduced to occupancy grids for planning navigation. This does not only safe space, but it is also crucial in providing object oriented access to environmental data as it is necessary in high-level applications such as map-based navigation.

To align map and environment, that is to match the map's features with those detected in the environment, we first have to interpret laser range finder data as shape data.

2.1 Interpreting Laser Scans as Shape Information

Laser range finders provide directional distance information to obstacles. This means they provide shape information about passable space. As suitable shape representation polygonal lines may be used, which connect detected reflection points. However, two aspects have to be considered:

- No information is available between two adjacent reflection points.
- Only obstacles in the line of sight are perceived, further objects may be hidden in the shadow.

The first aspect luckily needs no closer examination, as a laser range finder typically provide 361 measurements per 180°. This results in a distance of roughly 4cm for neighbored reflection points on a 7m distant surface, precise enough for regular indoor applications. However, matching has to take into account, that new features may appear when approaching an object. For example gaps between adjacent cupboards may only be detected from a head-on position. The second aspect is dealt with by dividing the perceived shape in partial shapes. This is done by removing lines nearly collinear with the line of sight if they are longer than a given threshold (e.g. 10cm), because these lines can block the sight. See figure 2 for an example on dividing shapes in scan data. The necessity of dealing with partial shapes imposes constraints on the applicability of shape matching techniques. Basically, only contour based methods have been proven to allow for dealing adequately with partial shapes [12].

2.2 Extracting Features by Shape Simplification

As mentioned above, schematic maps are rather imprecise. The geometric properties like exact angles or distances are of no importance, since only the presence of certain features at roughly given positions counts. In order to focus shape matching on these features, we apply shape simplification techniques to both, schematic map and perceived data. A suitable way to simplify a given shape is to use the discrete

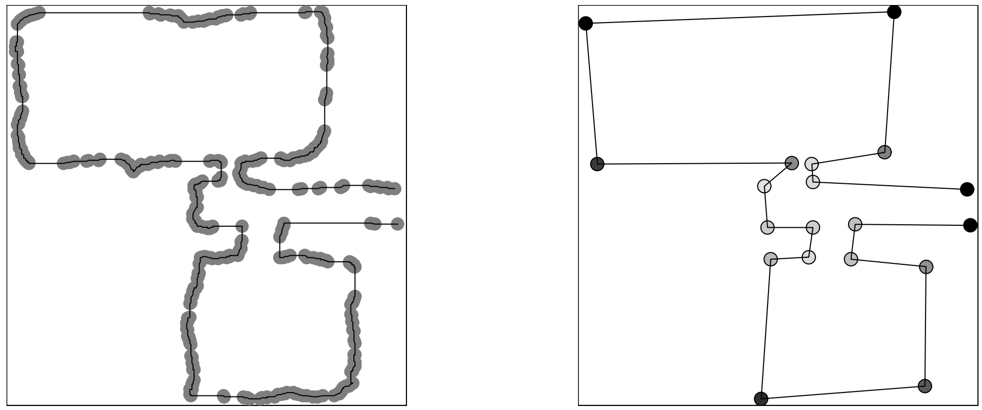

Figure 1: Shape preprocessing: The irrelevant and perhaps even faulty shape features of the original schematic map (left) are removed by discrete curve evolution (right).

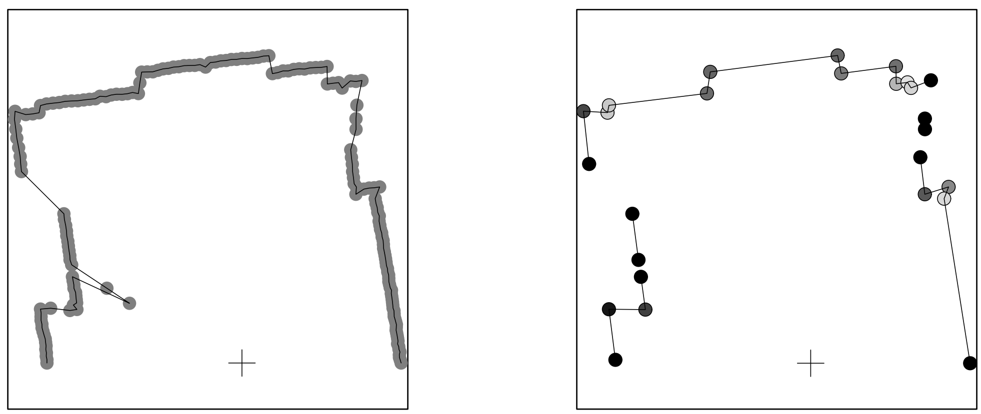

Figure 2: Shape preprocessing: The original scan (left, 181 scans per 180°) has been split into partial shapes. These shapes have been simplified (right). The vertices' color corresponds for non-end-points to its relevance measurement (see equation 1). Darker colors indicate higher relevance.

curve evolution proposed in [7]. This method may not only be applied to shapes but also to polygonal lines. According to [7], each vertex, which is not an end-point of a given polyline, is assigned a relevance measure given by:

$$K(s_1, s_2) = \frac{\beta(s_1, s_2) l(s_1) l(s_2)}{l(s_1) + l(s_2)}, \tag{1}$$

where s_1, s_2 are adjacent segments, $l(s_i)$ denotes a segment's length, and $\beta(s_1, s_2)$ stands for the enclosed angle. Starting with the vertex of lowest relevance, any vertex whose relevance is below a given threshold is deleted and the relevance measures of its neighbors are updated. The choice of a specific threshold is not critical [7]. An illustration of the shape preprocessing involved is illustrated in figure 1& 2.

3 Applying Shape Matching Techniques

To process the involved shape representations, we apply shape matching techniques originally developed for computer vision processes. The designed technique is derived from promising approaches proposed by Basri et al. [2] and Latecki & Lakämper [7]. Whereas the first approach is based upon a dynamic matching of line segments of two given contours, the latter computes a correspondence of maximal

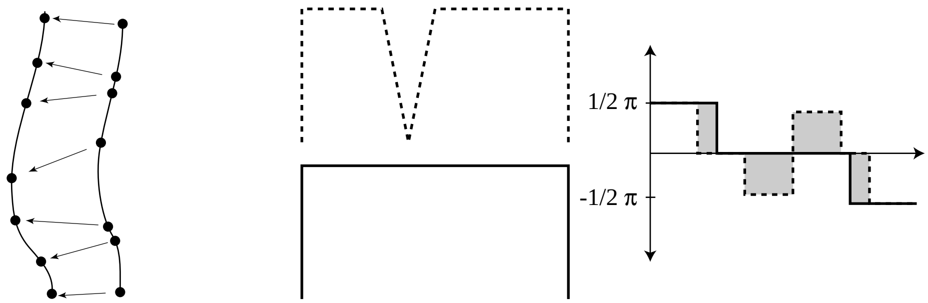

Figure 3: Left: Computing correspondences by matching vertices can cause unnecessary stretching. Similarity will then be underestimated. Right: Comparing arcs by integrating the difference in orientation may also lead to an underestimated similarity measure as local changes have global effects.

convex and concave arcs. Applied directly, neither of them seems adequate and promising in our domain. However, both techniques may be combined easily to achieve good results in our domain.

The disadvantage in using Basri's approach here directly lies in the combination of computing an optimal line segment correspondence according to an arc similarity measure based upon their physically motivated spring model. Each line segment is modeled as a spring which can be deformed by stretching, compressing, or bending. The dissimilarity of two arcs is given by the physical energy needed to deform the first arc into the second. Given two similar arcs with mainly different positions of vertices on the contour any matching requires many line segments to be stretched or compressed. Hence, the similarity is underestimated. An illustration of that problem is given in figure 3. The original work overcomes this problem by a specific smoothening of the contour not explained in full detail.

Latecki's work, though avoiding the disadvantages mentioned above by matching larger entities than simple line segments, has also minor problems. Here the line segments are grouped to maximal arcs which then serve as basic entities for the matching process. The similarity of two corresponding sequences of arcs is defined by a rather simple formula as integral of difference in orientation:

$$S(c,d) = \left(\int_0^1 (T(c)(s) - T(d)(s) + \Theta_{c,d})^2 ds\right) \max\{l(c), l(d)\} \max\{\frac{l(c)}{l(d)}, \frac{l(d)}{l(c)}\}, \tag{2}$$

where $T(c)(s)$ denotes the tangent direction of arc c at relative position s and $\Theta_{c,d}$ is the optimal rotation of the two arcs c, d minimizing the integral, see [8] for details. Using this method on simple shapes lacking a rich structure of arcs, the computed similarity relies basically on the arc difference measurement. Modifying an arc, for example by adding a peak, will drastically decrease its similarity with the unmodified arc. This is due to changes in corresponding points' relative position on the contour. The example of two adjacent cupboards with a suddenly appearing gap was mentioned above and illustrates this problem. A depiction is shown in figure 3. Unfortunately, indoor environments (especially offices) typically lack a rich shape structure.

The solution proposed here is rather simple but effective. We utilize Latecki's approach for matching maximal arcs, but replace the simple arc similarity. To compute the similarity of two given arcs we apply a dynamic matching of line segments like Basri et al.. Instead of using the spring model to determine dissimilarity, Latecki's arc similarity is applied here. The result is compared with the simple arc similarity measure given by 2, taking the minimum as determined similarity. This allows to overcome the problems mentioned above.

3.1 Matching Partial Shapes

So far, we have discussed the matching of two given polylines. However, in perceptual data only parts of shapes are visible. To account for this projection filters are applied [4] in order to determine the visible parts of the environment on basis of the estimated pose. Though, the example of the gap between two adjacent cupboards would confront us with problems again. So, instead of applying this approach to our domain, we attack this problem in a different way. Because we are dealing with objects instead of simple reflection points, the solution comes at hand. For each pair of contour to be matched we discard an arbitrary number of vertices from the beginning or the end. For matching two contours with n (rsp. m) vertices this involves $O(nm)$ partial shapes to be matched. As the number of vertices is usually low due to shape simplification and only local maps containing only few shapes are built, this rise of complexity is not significant. A small detail should not be left unmentioned: due to simplified shapes, discarding a whole line segment, which may be arbitrarily long, may discard just too much information. So the case that the first (rsp. last) line segment is only partially occluded is treated separately.

3.2 Respecting Positions

It is important to respect object positions in map-building, even when building only local perception maps. As objects at different positions may be indistinguishable by their features, shape matching like presented above is not sufficient for matching scans. Luckily, it can be extended easily by introducing a displacement error. Every polyline in the scan corresponding to a polyline in the local map determines the current pose, that is position and heading of the robot from which the scan must have been taken. As the robot can estimate its pose by means of odometry, a displacement error computed from the difference between estimated pose and the pose induced by the projection is added to the shape similarity measure. Computing scan alignments allows also to correct the pose estimated by the robot. Hence, only relative odometry is needed which is much more reliable than absolute one. The difference in pose can be computed by two corresponding pairs of vertices. Like [9] we compute orientation ω and translation T, averaging over the correspondences found. As our robot estimates its position much better than its orientation we use as dissimilarity between two poses $E(\Delta\omega, \Delta T) = \alpha||\Delta T||^2$.

3.3 Shape Matching for Scan Accumulation

As perception maps accumulate shape information to allow for model-based computation of shape correspondences, it is natural to apply shape matching techniques right away. As mentioned by Lu and Milios, "scan matching is similar to model-based shape matching" [9]. Due to noisy data and having no adequate way of dealing with imperfect models at hand their consequence has been to proceed in a different way. Since then, the field of shape representation has evolved drastically and inspired the approach presented here. Furthermore, not utilizing shape matching techniques for building local perception maps, we would employ the use of complicated and slow computer vision algorithms to extract shape information from the resulting data in order to align map and environment.

Accumulating scans to a perception map will be done in a straight forward way in spite of the problems known to such approaches like aligning scans which belong to a closed loop, etc.. As we are building only local perception maps, we are not confronted with the problems of a straight forward approach. Hence, this is a sufficient and efficient way. Now, to enter a scan in a partially built perception map, we need to:

1. Scan the scene and apply shape processing as described in section 2.2 to cancel noise. This results in some only slightly simplified partial shapes.
2. Try to match every perceived partial shape with the map respecting pose and visibility. Besides discarding vertices as described in section 3.1 the latter will be accounted for by considering only matches with a dissimilarity lower than a given threshold.
3. Correct the robots pose on basis of correspondences detected.
4. Update the map. That is, enter every partial shape which does not belong to a correspondence; these are the parts of environment that just came into sight. The partial shapes belonging to correspondences may be used to improve the map. We simply replace the ones from the map with the newly scanned ones, as this is sufficient for our task.

3.4 Aligning Map and Environment

To align map and environment, the described shape similarity is utilized again. High shape similarity between partial shapes in perceptual and schematic map point to good candidates. So aligning map and environment has become much like building the local perception map. However, there are some points to consider:

- the schematics map's coarse granularity,
- no good pose estimation may be available, and
- the map's scale is unknown.

Shape simplification by means of discrete curve evolution provides us with the method to overcome the first problem. So, shapes are simplified to a high degree so that only important features, which should be represented in a helpful schematic map remain. The influence of pose estimation and consequently pose difference is controlled by the displacement error $E(\Delta\omega, \Delta T) = \alpha||\Delta T||^2$. (see section 3.2) we can account for less certainty of pose estimation by using a lower value for α. Coping with unknown scale is no more difficult, as the scale's influence on shape similarity can be neglected by normalizing curve lengths prior to comparison.

4 Map-based Navigation

As the schematic map provided is not only instruction but also a navigation aid available from the very beginning of the task, the robot should use it to best advantage. Therefore, no exploration and map building phase should be required before carrying out the task. This can be achieved by decomposing the given map in cells that can be recognized in the environment because of their shape features. So the task can be executed iteratively on the basis of sub-goals.

4.1 Cell Decomposition to allow for Non-Metric Navigation

The schematic map is partitioned into meaningful regions relevant to both, navigation and self localization. Accordingly, regions are merged, which are neighbored in voronoi-like diagrams. Again, we will take advantage of shape simplification as this yields more stable and far more simple diagrams. Typically, Kuipers' places [6] are located on these cell's boundary. The idea behind our partition-based navigation strategy is the following: though these cells need not to be convex like in conventional decomposition approaches, robot motion planning within each cell is a rather simple task, as it is easy to get a view of a cell's region. Therefore, robot motion in cells can be based on low-level behaviors and can be separated from the navigation strategy. This navigation strategy does not need to take into account any detailed geometric features of cells. Hence, it is efficient and robust to noisy data.

The identification of one particular cell and the robot's position within it is to be detected by the cell's shape features and has been discussed in section 3.4. Sufficient information for navigation is provided by a connectivity structure with relative positions. So, cells need to be attributed with their neighbors determined by the existence of a directly connecting path in the plane.

4.2 Dual Navigation

The robot's route can be computed on the basis of the cell's topological structure by graph search algorithms, with respect to attributed cells (e.g. perimeter classification from small, medium-size, up to large) the problem is a shortest path problem in a weighted graph. Navigation is performed on a cell-to-cell basis. A high-level planner invokes cell per cell on the route as next goal, which is passed on to a locomotion module. For navigating to the next goal cell, low-level motion behaviors are sufficient. Self localization essential in map-based navigation is also divided into two separated aspects: identifying the presence of the next cell on the one hand and localizing the robot within on the other hand. Therefore, there is no need to accumulate a global metric map.

5 Conclusion

This approach provides a different view on map-aided robot navigation. It does not require building a global metric perception map. Since global accumulation suffers from local inaccuracies summing up into global inconsistencies, special care would have to be taken, e.g. to prevent disjunctive areas to overlap (e.g. two rooms separated by a wall), or to align local maps belonging to a closed loop. This approach avoids these problems as there are only local perception maps which can be built much easier in a straight forward way. The local perception maps used are made of polygonal lines which can be viewed as shape features. Applying shape matching techniques to the problem of aligning multiple range finder views or aligning perception and schematic map is a promising approach. Navigation is divided in low-level motion behaviors and qualitative route planning. For higher levels of navigation we use higher abstractions in our spatial representation. Topological information attributed with qualitative spatial information about directions and distances is well-suited for performing navigational tasks with schematic maps.

So far, using contour based shape representations has one main disadvantage. They do not represent the passable space itself but only its outline, so scan-matching algorithms aligning only the contour may align a contour with a part from the map hidden behind another object and therefore not visible from the current viewpoint. Future work will focus on different shape representations which will allow us to account for that. A closer relation between the shape representation used for map related processes and the decomposition and its construction is intended. An adequate shape representation may bridge the gap between sub-symbolic, metric information needed and more abstract, qualitative information desired. This motivates further research.

As author I would like to thank my colleagues for numerous discussions and helpful remarks, of them especially Prof. Christian Freksa, Jan Oliver Wallgrün, and Sven Bertel.

References

[1] Barkowsky, T., Latecki, L. J., and Richter, K.-F.: Schematizing Maps: Simplification of Geographic Shape by Discrete Curve Evolution, In: Freksa, Brauer, Habel, Wender (eds): Spatial Cognition II Integrating Abstract Theories, Empirical Studies, Formal Methods, and Practical Applications, Springer-Verlag, 2000

[2] Basri, R., Costa, L., Geiger, D., and Jacobs, D.: Determining the similarity of deformable shapes, Vision Research, 38, pp 2365–2385, 1998

[3] Freksa, C., Moratz, R., and Barkowsky, T.: Schematic Maps for Robot Navigation, In: Freksa, Brauer, Habel, Wender (eds): Spatial Cognition II Integrating Abstract Theories, Empirical Studies, Formal Methods, and Practical Applications, Springer-Verlag, 2000

[4] Gutmann, J.-S. and Nebel, B.: Navigation mobiler Roboter mit Laserscans, Autonome Mobile Systeme (AMS'97), Springer-Verlag, 1997

[5] Gutmann, J.-S. and Schlegel, C.: AMOS: Comparison of Scan Matching Approaches for Self-Localization in Indoor Environments, Proceedings of the First Euromicro Workshop on Advanced Mobile Robots EUROBOT '96), October 1996, pp 61–68

[6] Kuipers, B.: The Spatial Semantic Hierarchy, Artificial Intelligence, 119, pp 191–233, 2000

[7] Latecki, L. J. and Lakämper, R.: Shape decomposition and shape similarity measure. In Proceedings of the 20. DAGM-Symposium Mustererkennung (Pattern Recognition), Stuttgart, pp 367–376. Springer-Verlag, 1998

[8] Latecki, L. J. and R. Lakämper: Application of Planar Shape Comparison to Object Retrieval in Image Databases. Pattern Recognition (PR), to appear.

[9] Lu, F. and Milios E.E.: Robot Pose Estimation in Unknown Environments by Matching 2D Range Scans, Proceedings IEEE Conference on Computer Vision and Pattern Recognition, p 935–938, 1994

[10] Thrun, S.: Learning Metric-Topological Maps for Indoor Mobile Robot Navigation, Artificial Intelligence, 99, pp 21–71, 1998

[11] Weber, H.: Handwritten sketches as user interface in movement orders for mobile robots, In: Proceedings of the Third International ICSC Symposia on Intelligent Industrial Automation IIA'99 and Soft Computing SOCO'99, Genova, Italy, 1999, pp 119–126

[12] Wolter, D.: Formrepräsentation — Architekturvorschlag für Anwendungen in einem Robotikkontext, diploma thesis, University of Hamburg, 2000