# Shape-Based Robot Mapping

Diedrich Wolter[1], Longin J. Latecki[2], Rolf Lakämper[2], and Xinyui Sun[2]

[1] Universität Bremen
[2] Temple University

**Abstract.** We present a novel geometric model for robot mapping suited for robots equipped with a laser range finder. The geometric representation is based on shape. Cyclic ordered sets of polygonal lines are the underlying data structures. Specially adapted shape matching techniques originating from computer vision are applied to match range scan against the partially constructed map. Shape matching respects for a wider context than conventional scan matching approaches, allowing to disregard pose estimations. The described shape based approach is an improvement of the underlying geometric models of todays SLAM implementations. Moreover, using our object-centered approach allows for compact representations that are well-suited to bridge the gap from metric information needed in robot motion and path planning to more abstract, i.e. topological or qualitative spatial knowledge desired in complex navigational tasks or communication.

**Keywords:** cognitive robotics, robot mapping, shape matching

## 1 Motivation

The problems of self-localization, i.e. localizing the robot within its internal map, and robot mapping, i.e. constructing the internal map autonomously, are of high importance to the field of mobile robotics [18]. Coping with unknown or changing environments requires to carry out both tasks simultaneously, therefore this has been termed the SLAM problem: Simultaneous Localization and Mapping [5]—it has received considerable attention [5, 8, 18]. Successful stochastical approaches have been developed that tackle representation and handling of uncertain data which is one key point in SLAM. As todays stochastical models are powerful, even linking them to a simple geometric representation like reflection points measured by a range sensor already yields impressive results. Advances in stochastical means have improved the overall performance leaving the basic spatial representation untouched. As the internal geometric representation is a foundation for these sophisticated stochastical techniques, shortcomings on the level of geometric representation affect the overall performance.

We claim that an improved geometric representation enhances the overall performance dramatically. A compact, object oriented representation based on shape is an universal yet slender one. It can outperform often-used occupancy grids in storage as well as in computational resources, since smaller sets of data need to be processed. Object-centered representations have been judged necessary to represent dynamic environments [18]. Moreover, a more comprehensive spatial representation can allow to mediate between different aspects of spatial information that are desired or even necessary in applications. We propose a shape representation of the robot's surrounding that grants access to metric information as needed in robot motion or path planning alongside with more abstract, qualitative or topological knowledge which is desired in navigational tasks and a well-suited foundation for communication.

## 2 Related Work

Any approach to master the SLAM problem can be decomposed into two aspects: handling of map features (extraction from sensor data and matching against the (partially) existing map) and handling of uncertainty.

To address uncertainty mainly statistical techniques are used, e.g., particle filters, the extended Kalman filter, a linear recursive estimator for systems described by non-linear process models and/or observation models, are used in most current SLAM algorithms [17, 18, 8]. As this paper focusses exclusively on the map's geometric representation, we now review related aspects in detail.

Typically, map features extracted from sensor data (esp. range finder data) are either the positions of special landmarks [5], simple geometric features, especially lines [12, 13, 3], or range finder data is used rather directly [18].

Direct use of data, that is without further interpretation despite noise filtering, results in constructing a bitmap-like representation of the environment termed occupancy grid [6]. The simplicity of this approach causes its strength, namely universality: It may be used in unstructured, unprepared environments. However, harmful features show off as well. First, the crucial method of matching a scan against the map in order to localize the robot is formulated as a minimization [12, 18, 8]. Therefore, a good estimation of the robot's pose is required to prevent minimization getting stuck in local minima. Second, occupancy grids grow with the environment's size, not its complexity. As grids need to fine, using these maps can easily end up in handling large data sets. This is not only a problem of storage, but, far more important, it affects run-time of algorithms handling the map as huge amounts of data need to be processed. To keep path planning in a once constructed map feasible, a topological representation can be coupled with the metric one [16].

To maintain a map at manageable size from the beginning, representations based on features or landmarks provide excellent means. These so-called object maps represent only position of landmarks and their distinctive features. Thus, these maps grow with the environment's complexity (i.e. the number of landmarks visible). This allows for an efficient processing. Using natural landmarks is of special interest as environments do not need to be prepared, like, e.g., by installing beacons [5]. For example, mapping based on line segments has been shown to improve performance in office environments [13]. A key point in feature-based approaches is a matching of perceived features against the ones represented in the map. Wrong matchings result in incorrect, hence, useless maps. Complex features help to prevent a mixup when matching the robot's perception against its map. As features' presence is required, application is often limited to special environments only. On the contrary, choosing simple, omnipresent features can easily inhibit a reliable matching of perceived features against the map. Unreliable feature extraction, e.g. extracting line segments from round objects causes problems, too, as additional noise gets introduced. As noise gets propagated, it sums up and can cause inconsistent maps.

To overcome these problems, we propose a shape based representation that is (a) universal as employed shape features can be extracted in any environment, but (b) features provide distinctive information as shape respects a wide spatial context. Matching of features is, thus, based on a shape matching which has received much attention in the context of computer vision.

The idea of applying shape matching in the context of robot mapping is not new. In the fundamental paper by Lu & Milios [12], scan matching has already been considered similar to model-based shape matching. Thrun considers this connection underexploited [18]. Recent advances in shape matching provide a good starting point to bring these fields together. Our approach utilizes a model based similarity measure.

In the domain of robot mapping mainly two key aspects dictate to the applicability of shape descriptors, namely partial shape retrieval and the ability to deal with simple shapes. Firstly, as only partial observations of the environment can be made, any approach to shape representation that cannot handle partial shapes renders itself unemployable. This includes, for example, encoding by feature vectors like Fourier or momenta spectra. Secondly, any robot's working environment must be representable in the framework of the chosen shape descriptor. Besides these confinements, investigating into shape information available in typical indoor environments displays another feature required: Much shape information perceivable is often rather poor, like for instance straightaway walls with small protrusions only. Therefore, shape recognition processes must be very distinctive, even on rather featureless shapes.

Structural approaches represent shape as a colored graph representing metric data alongside configurational information. Amongst these so-called skeleton based techniques, especially shock graphs (cp. [15]) are worth consideration[3]. Though primarily structural approaches may very well bridge the gap from metric to more abstract qualitative or topological information (cp. [16]), recognizing shapes lacking of a rich structure of configuration, has not yet proven feasible. Moreover, a robust computation and matching of a skeleton in the presence of noise has not yet been solved.

Therefore, we propose the utilization of a boundary based approach. Considering the discrete structure provided by sensors, using polygonal lines to represent the boundaries of obstacles may be achieved easily. Related techniques for matching rely on a so-called similarity measure. Various measures, often metrics, have been developed. Arkin et al. ([1]) accumulates differences in turning angle in straightforward manner; it fails to account for noise, esp. if not uniformly

[3] Skeleton based approaches relate closely to Voronoi based spatial representations used in the field of robotics (cp. [16, 15]).

distributed, adequately. Basically all improvements in similarity measures, thus, employ a matching of boundaries to establish a correspondence prior to summing up dissimilarities of corresponding parts. Basri et al. propose a physically motivated deformation energy ([2]). More recently, an alignment-based deformation measure has been proposed by Sebastian et al. which considers the process of transforming one outline into another ([14]). However, common to these approaches is that an equal sampling rate of the outlines is required. The emerging problem of comparing two, let us assume identical outlines with different sample points, can be illustrated easily: Computing a correspondence of either points or line-segments introduces large deformations due to mismatches in point correspondence, hence, underestimating similarity dramatically. Considering shape information obtained by a range sensor, scanning the same object from different positions generates this effect.

An improved performance in similarity measures for closed contours has been achieved by Latecki & Lakämper who consider a matching on basis of a a priori decomposition into maximal arcs (cp. [10]). We will formulate the presented approach on this basis. However, it is adapted such that it is tailored to deal with any kind of open polyline and addresses the problem of noisy data in a direct manner. The representation is complemented by a structural representation of robust ordering information. Applicability of the elementary shape similarity measure has been shown in [11].

## 3 Structural Shape Representation

Shape information can directly be derived from sensor readings obtained by a range sensor, typically a laser range finder (LRF). Shape is represented as a structure of boundaries. Polygonal lines, called *polylines*, serve as the basic entity.

Polylines represent obstacles' boundaries. Much of the spatial information represented in the map can be captured by individual polylines which form visual parts (cp. [10]). The variety of perceivable shapes in a regular indoor scenario already yields a more reliable matching than other feature-based approaches, as mixups in determining corresponding features are more unlikely to occur. At the same time, we are able to construct a compact representation for an arbitrary environment. However, we exploit even more context information than represented by a single polyline considering shape as a structure of polylines. This allows us with basically no extra effort to cope with environments displaying mostly simple shapes. The structure captured is ordering information. For any given viewpoint, perceivable objects can be ordered in a counter-clockwise manner. Thus, for a map containing polylines the structure of ordering can be expressed as a mapping from a point (the robot's position in the map) to a vector of polylines. Given a polygonal map, computing the vector of visible polylines can be achieved by applying a sweep line algorithm used in computational geometry to determine visibility [4]. A first step in the presented approach, however, is to extract shape information, i.e. polylines from data acquired by the LRF.

### 3.1 Grouping and Simplification of Polylines

Let us assume that the range data is mapped to locations of reflection points in the Euclidean plane, using a local coordinate system. Now, these points are segmented into individual polylines which represent visual parts of the scan. For this grouping a simple heuristic may be employed: Traversing the reflection points in the (cyclic) order as measured by the LRF, an object transition is said to be present wherever two consecutive points are farer apart than a given distance threshold. For obtaining the experimental results we used a threshold of 20cm, however the precise choice is not important, as differences in grouping are accounted for (cp. section 4.2).

Polylines extracted this way still carry all the information (and noise) retrieved by the sensor. To make the representation more compact and to cancel out noise, we employ a technique called Discrete Curve Evolution (DCE) introduced by Latecki & Lakämper ([9]) to (a) make the data more compact without loosing valuable shape information and (b) to cancel out noise. DCE is a context-sensitive process that proceeds iteratively: *Irrelevant* vertices get removed until no irrelevant ones remain. Though the process is context-sensitive, it is based on a local relevance measure for a vertex $v$ and its two neighbor vertices $u, w$[4]:

$$K(u, v, w) = |d(u, v) + d(v, w) - d(u, w)| \tag{1}$$

Hereby, $d$ denotes the Euclidean distance. The process of DCE is very simple and proceeds in straightforward manner. The least relevant vertex is removed until least relevance exceeds a given simplification threshold. Proceeding this way we obtain a cyclic ordered vector of polylines. Consequently, as no relevance measure is assigned to end-points, they remain fixed. The choice of a specific simplification threshold is not crucial, since only the overall shape needs to be preserved. The precise classification of noise will done in the context of corresponding polylines (cp. section 4.1). Exemplary results for applying DCE to LRF data are shown in Figure 1; Figure 2 demonstrates suitability for curved boundaries.

## 4 Matching of Shapes

To match two shapes means to match two ordered set of polylines against each other. Whereas one shape has been extracted from a sensor reading, the other is determined by the partially built map. Based on an estimation of the robot's position in its internal map, the shape perceivable according to the map is computed. To localize the robot and update its map, visual parts perceived by the sensor need to be matched against those extracted from the map. Hence, we need to seek for the *best* correspondence of individual polylines that preserves the shapes' structure, i.e. which does not violate their order. Shape similarity is the key point to quantify quality of a correspondence.

[4] Context is respected as in the course of simplification the vertices' neighborhood changes.

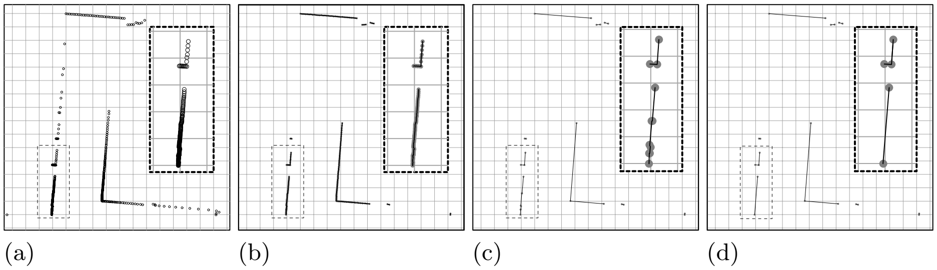

**Fig. 1.** Extracting polylines from a scan. Raw scan points (a) are grouped to polylines (b), then simplified by means of DCE. The threshold used in (c) is 1.0 and 5.0 in (d). The two additional rectangles show magnifications of marked parts. The grid denotes 1 meter distance.

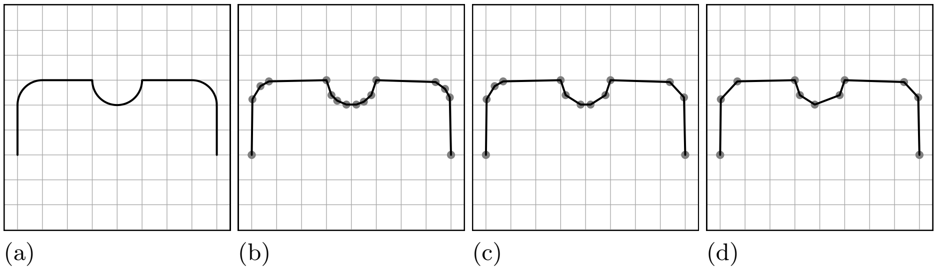

**Fig. 2.** Extracting shape information from curved synthetic data by DCE. The original data is shown in (a), (b) displays the result using 1.0 as stop threshold. A threshold of 2.5 was used in (c) and 5.0 in (d) (cp. Figure 1 for real LRF data). The grid denotes 1 meter distance.

### 4.1 Similarity of Polylines

The similarity measure utilized in our approach is based on a measure introduced by Latecki & Lakämper which we first will summarize very briefly and indicate changes made as it is necessary in this context; for details refer to [10].

To compute the basic similarity measure between two polygonal curves, we establish the best possible correspondence of maximal left- or right-arcuated arcs[5]. To achieve this, we first decompose the polygonal curves into maximal subarcs which are likewise bent. Since a simple one-to-one comparison of maximal arcs of two polylines is of little use, due to the fact that the curves may consist of a different number of such arcs and even similar shapes may have different small features, we allow for 1-to-1, 1-to-many, and many-to-1 correspondences of maximal arcs. The main idea here is that we have at least on one of the contours a maximal arc that corresponds to a part of the other contour composed of adjacent maximal arcs. The best correspondence can be computed using Dynamic Programming, where the similarity of the corresponding visual

[5] The original work is based on convex and concave arcs respectively. As we deal with open polylines here, the terms convex or concave would be meaningless.

parts is as defined below. The similarity induced from the optimal correspondence of polylines $C$ and $D$ will be denoted $S(C, D)$.

Basic similarity of arcs is defined in tangent space, a multi-valued step function representing angular directions and relative lengths of line-segments only. It was previously used in computer vision, in particular, in [1]. Denoting the mapping function by $T$, the similarity gets defined as follows:

$$S_a(C, D) = (1 + (l(C) - l(D))^2) \cdot \int_0^1 (T_C(s) - T_D(s) + \Theta_{C,D})^2 ds \qquad (2)$$

where $l(C)$ denotes the arc length of $C$. The constant $\Theta_{C,D}$ is chosen to minimize the integral (cp. [10]) (it respects for different orientation of curves) and is given by

$$\Theta_{C,D} = \int_0^1 T_C(s) - T_D(s) ds. \qquad (3)$$

Obviously, the similarity measure is a rather a dissimilarity measure as the identical curves yield 0, the lowest possible measure. This measure differs from the original work in that it is affected by an absolute change of size rather than a relative one. It should be noted that this measure is based on shape information only, neither the arcs' position nor orientation are considered. This is possible due to the large context information of polylines.

A problem of comparing polylines extracted from LRF data is that often the amount of noise and the size of shape features present is challenging. Applying DCE to a degree that would certainly remove all noise would remove many valuable shape features as well. DCE makes vertex removal decisions in the context of a single object. A better noise identification can made in the context of comparing corresponding polylines. Therefore, we encapsulate the basic similarity measure $S$ in another process that masks out noise in the context of corresponding polylines. It is similar to the initial curve evolution employed. When comparing a polyline $C$ perceived by the sensor and a polyline $D$ extracted from the map, $C$ might still contain extra vertices caused by noise. Therefore, we continue evolving polyline $C$ if the resulting similarity measure improves. The maximal similarity obtained this way, i.e. the lowest value of $S(C, D)$, is denoted $S^\star(C, D)$. To make this process robust against local minima, a small lookahead is used. This means that evolution is continued as long as removing the next 3 vertices according to the simplification rule yields a gain in similarity. The reason for delaying the final curve evolution steps to the matching is to enable the exploitation of even small shape features. Already starting the DCE process on sensor data, on the contrary, allows to benefit from DCE's lower computational complexity. Proceeding this way, only a few simplification steps need to be carried out during the matching. An example is depicted in Figure 3. To enhance presentation, the preceding DCE has been left out. When comparing the two polylines shown in Figure 3 (a) and (b), vertices from the perceived contour (b) are removed in the order of vertex (ir-)relevancy while the shape similarity improves. The similarity values in the course of the evolution and the resulting, simplified polyline are shown.

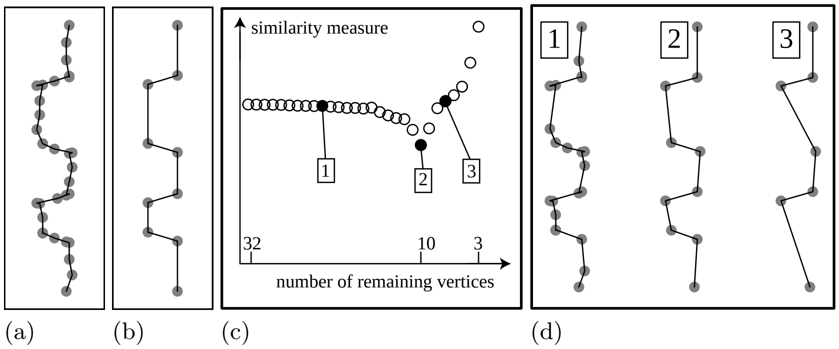

**Fig. 3.** Computation of the model based similarity measure. A perceived, distorted polyline (a) is compared with one extracted from the robot's internal map (b). Curve evolution of distorted polyline is continued as long as the similarity between the two polylines improves. The development of similarity values in the course of the simplification is given in (c), the X axis represents simplification stages (decreasing number of remaining vertices). The evolution would stop at the position marked 2, the best similarity. (d) Different stages in the evolution process marked in (c). The polyline resulting from the model based evolution is marked 2.

### 4.2 Matching of Polylines

Computing the actual matching of two structural shape representations extracted from scan and map is performed by finding the *best* correspondence of polylines which respects the cyclic order. For the ease of description, let us assume that no information about the robot's movement since the last matching is available. Consequently, the shape is extracted from the map according to the last view point determined (cp. 5.1). Shape similarity is the key to measure quality of a matching. Additionally, we must take into account that (a) not all polylines may get matched as features' visibility changes and (b) that due to grouping differences (cp. section 3.1) not necessarily 1-to-1 correspondences exist. Noise or change of view point, for example, may lead to a different grouping. Moreover, since every correspondence of polylines induces an alignment that would align scan and map, we demand all induced alignments to be very similar. This criterion is helpful to correctly match featureless shapes, e.g. short segments like obtained when scanning a chairs' legs. The clue in our approach is exploiting correspondence of salient visual parts to correctly identify featureless ones even if no a priori alignment is available (cp. [8]).

An estimation of the alignment, or equivalently: the robot's position in the internal map, is necessary to utilize an efficient matching algorithm. We will show in (Section 4.3) how to compute it using shape similarity. Clearly, it can be derived from odometry if odometry data is available.

Let us now assume that such an estimation exists. Let us further assume that $\boldsymbol{B} = (B_1, B_2, \ldots, B_b)$ and $\boldsymbol{B'} = (B'_1, B'_2, \ldots, B'_{b'})$ are two cyclic ordered vectors of polylines. Denoting correspondence of $B_i$ and $B_j'$[6] by relation $\sim$, the task can formulated as minimization.

$$\sum_{(\boldsymbol{B}_i,\boldsymbol{B'}_j)\in\sim} (S^\star(\boldsymbol{B}_i,\boldsymbol{B'}_j) + D(\boldsymbol{B}_i,\boldsymbol{B'}_j)) + \sum_{B\in\tilde{B}} P(B) + \sum_{B'\in\tilde{B}'} P(B') \overset{!}{=} \min \quad (4)$$

Hereby, $\tilde{B}$ (rsp. $\tilde{B}'$) denotes the set of polylines not belonging to any matching. $P$ denotes a penalty function for not matching a polyline. This is necessary, as not establishing any correspondence would otherwise yield the lowest possible similarity 0. The penalty function is chosen to linearly grow with the polyline's size modeling a higher likelihood for smaller polylines to appear or disappear[7]. $D$ denotes the aforementioned alignment measure quantifying the deviation of the estimated alignment from the one induced by the correspondence $\boldsymbol{B}_i \sim \boldsymbol{B}_j'$. The best correspondence can so be computed by applying an extended Dynamic Programming scheme. The extension regards the ability to detect 1-to-many and many-to-1 correspondences of polylines. The basic idea here is to consider in each step of the computation if it is advantageous to establish a grouping with the latest correspondence determined so far, i.e. if the summed up (dis-)similarity values and skipping penalties can be decreased. This results in a linear extra effort such that the overall complexity for matching two vectors of $n$ polylines each is $O(n^3)$—low enough that our prototypical implementation on a standard computer can process several scans per second.

### 4.3 Matching in the Absence of Odometry

The outlined matching is already capable of tracking complex shapes even if no estimate of the induced alignment is available, because shape similarity is very distinctive. We will detail now on obtaining an alignment estimation purely by shape similarity.

If we had two corresponding polylines, hence, the induced alignment, we could use this as the estimation in the matching. Observing that many shapes can be matched only in consideration of shape similarity, the matching can be employed to obtained this correspondence[8] Thus, the matching can be computed in a two pass process. Within the first matching pass the consideration of induced alignments' similarity is ineffective. Then, the *most reliable* correspondence is selected. Finally, the actual matching is computed using the alignment induced by the selected matching. To quantify reliability, a measure based on shape similarity and shape complexity has been proposed [11]. A polyline's shape complexity may be expressed by summing up inner points' relevance measures

[6] To be more precise: correspondences of either $B_i$ and $\{B_j', B_{j+1}', \ldots, B_{j'}'\}$ or $\{B_i, B_{i+1}', \ldots, B_{i'}'\}$ and $B_j'$ since we consider correspondences of types 1-to-many and many-to-1, too.

[7] When comparing likewise noisy polylines, similarity values grow linearly with the polylines' size, too.

[8] As there are not necessarily 1-to-1 correspondences, it might not be sufficient to only consider individual similarities.

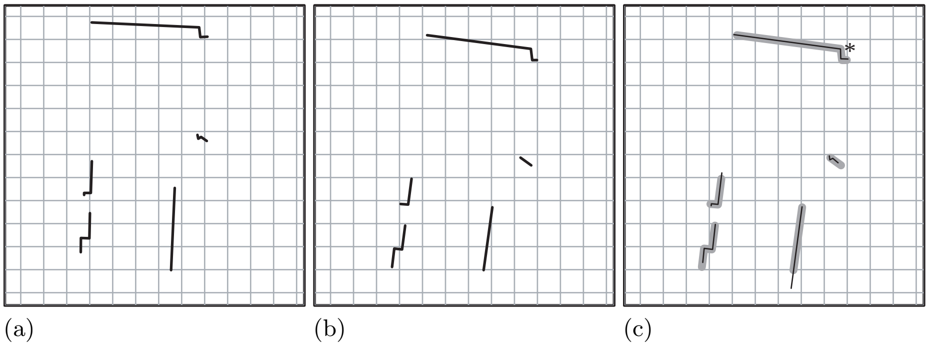

**Fig. 4.** Exemplary results of the shape based matching. Shape perceived by the LRF (a) is matched against the shape extracted from the map (b). Shapes are matched in a two step process. In the first step, only the most reliable matching (marked $\star$ in (c)) is used to align perception and map. In the second step, the final matching is computed and the perception is aligned. (c) Shows the alignment of (a) and (b), using thick lines for the map shape and thin lines for the perceived shape. The grid in the illustrations denotes 1m distance.

(cp. equation 1). If a polyline has no inner points, complexity is given by half the Euclidean distance of its end points. Terming this complexity measure $C$, the reliability is defined as

$$Q(P,Q) = C(P) + C(Q) - S^{\star}(P,Q). \tag{5}$$

The idea is expressing reliability as high similarity of complex shapes (cp. [11] for details). An exemplary result is presented in Figure 4 where a scan is matched against the shape extracted from the map (a). Based on the most reliable correspondence the estimated alignment, i.e. the position of the robot within the map, is computed. Accordingly aligned scan and map excerpt is shown together with the computed matching in (b). The presented technique can cope with differences in position estimates of more than 1m; rotational changes that retain the visibility of most salient objects are mastered, too. Observe, that this is a dramatical improvement compared to the precision required by standard scan matching approaches which typically rely on a hill climbing strategy (cp. [8]).

## 5 Map Update

Once the shape extracted from the map and the shape perceived by the sensor have been matched, the internal map can be updated. The first step is to align perceived shape and map. Next, corresponding polylines can be merged to obtain a single polyline in the map comprising existing map and new shape information. Perceived polylines not corresponding to a polyline in the map are considered to be a newly emerged features. Thus, they are added to the map.

### 5.1 Alignment & Localization

To localize a robot within its map, the perceived scan needs to be aligned with the internal map. This yields the position of scan's origin in the map's frame of reference, hence, provides the localization. To align perceived scan and map, we adapt a scan matching technique originally developed by Cox [3] and improved by Gutmann [7].

The adapted approach is based on a scan point to line matching, i.e. reflection points measured by the LRF are matched against model lines represented in the map. Distance of scan points and model lines is minimized by aligning the scan. Given a correspondence of points and lines, the optimal alignment, i.e. a rotation around the scan's origin and a translation, can be computed in a closed form (cp. [7]). The correspondence, however, might not be correct from the beginning, since it is based on rather simple rule. The idea to overcome this problem is carrying out several steps of matching and aligning iteratively, to allow convergence of correspondences and, thus, alignment. Limitations of this procedure are due to the rather simple matching rule which considers no large spatial context than a simple line. A good a priori alignment are therefore a prerequisite for successful operation. Even with small distortions, some scan points may get matched to the wrong model line. This can cause the process to get trapped in a local minima.

To apply the method of Cox, we need to improve the matching of points and model lines. The knowledge of matching shapes is the key here. Only correspondences between corresponding shapes are considered in the alignment process. For each perceived shape sample points are determined. To obtain the experimental results in this paper, we used a sampling of 10 cm. Additionally, we ensured that at least one sampling point lies on each of the polyline's line segments. Using these sampling points, corresponding line segments of corresponding shapes are determined based on proximity. For every sampling point the nearest point contained in the corresponding polyline is computed and the scan is aligned. The procedure is repeated until convergence is reached, i.e. the alignment does not change significantly any more. Experiments show that this rather simple, straightforward adaption already yields good results.

### 5.2 Merging of Polylines

After perceived shape and map have been aligned, we need to merge corresponding polylines to obtain a single, comprehensive one. This way, newly detected features can be added to a polyline. Moreover, it can be mediated between differing perceptions of the same polyline. We need to account that due to change of visibility only some parts of the polylines may actually correspond. Hence, we first decompose each polyline into *head*, *body*, and *tail* (cp. Figure 5). The body part refers to the corresponding part of the polyline. Head and tail denote the remaining parts.

To obtain the merged polylines, the merged body parts are appended with head and tail parts. Note that there exists at most one head and one tail. To

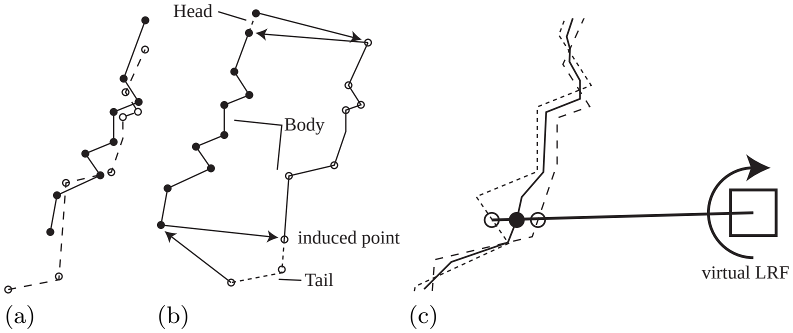

**Fig. 5.** Illustration of the merging process. (a) Two aligned polylines (solid and dashed lines) are the starting point. (b) End points are mapped to nearest points on the other polyline (For illustration purposes the lines have been shifted apart). The nearest points induced determine beginning of head and tail parts. (c) To average the corresponding body parts, a laser scanner is simulated. The solid line shows the result of averaging the two dashed lines which show a magnified excerpt from the body parts shown in (b).

determine the body parts of corresponding polylines, the shortest distance between a polyline's end points and the other polyline are computed. This induced two additional points on the other polyline. The body parts are said to be determined by the points whose nearest points not coincide with an end point. Refer to Figure 5 for illustration.

To merge the body parts, we use a simple technique. A laser range finder located at the robot's current position in the map is simulated. It scans both body parts simultaneously. Averaging the distances measured yields a new, comprehensive body part. To obtain a more sophisticated mediation between newly perceived shape and map, a weighted average can be used. For example, as a polyline in the map may result from many measurements, a single new measurement should not change the polyline dramatically any more. Since this closely relates to handling of uncertainty and stochastical models which have been masked out in this paper, this issue is not detailed any further here.

## 6 Experimental Results

In our experiments, we have processed data obtained from SICK LMS laser range finders mounted on a Pioneer-2 robot or on the Bremen autonomous wheelchair [13]. Figure 6 shows the resulting map from processing 450 scans taken at a rate of aprox. 15 scans per second. The average robot speed was 0.5 m/sec. No odometry information has been used to obtain the results.

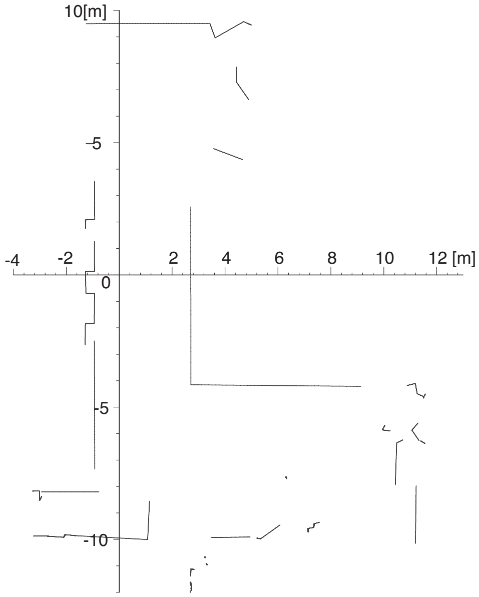

**Fig. 6.** A shape based map consisting of only a few polygonal lines. It has been constructed autonomously from processing 450 laser range finder scans recorded at University of Bremen. The unit size is 1m.

## 7 Conclusion and Outlook

We have presented a comprehensive geometric model for robot mapping based on shape information. A shape matching has been specially tailored to match shape perceived by a laser range finder against a partially existing polygonal map. The matching is powerful enough to disregard any pose information and cope with scans significantly differing from the map. Based on corresponding shapes, the process of localization and map update have been detailed.

We are aware that statistical methods are needed to guarantee robust performance, but did not include any as we concentrated on geometric models exclusively. So, future work comprises the coupling with a state-of-the-art stochastical model. Since object centered approaches are judged well-suited to handle dynamics and shape matching enables us to identify corresponding objects, explicit handling of dynamics within our architecture is of great interest to us. Additionally, as we believe that our shape based approach is particularly promising in attacking the problem of cycle detection, we plan to investigate into this topic.

## Acknowledgment

This work was carried out in the framework of the SFB/TR 8 Spatial Cognition, project R3 [Q-Shape]. Financial support by the Deutsche Forschungsgemeinschaft is gratefully acknowledged. Additionally, this work was supported in part by the National Science Foundation under grant INT-0331786 and the grant 16 1811 705 from Temple University Office of the Vice President for Research and Graduate Studies. Thomas Röfer is acknowledged for providing scan data.

## References

1. M. Arkin, L.P. Chew, D.P. Huttenlocher, K. Kedem, and J. S. B. Mitchell. An efficiently computable metric for comparing polygonal shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13, 1991.
2. Ronen Basri, Luiz Costa, Davi Geiger, and David Jacobs. Determining the similarity of deformable shapes. *Vision Research*, 38, 1998.
3. Ingemar J. Cox. Blanche: Position estimation for an autonomous robot vehicle. In Ingemar J. Cox and G.T. Wilfong, editors, *Autonomous Robot Vehicles*, pages 221–228. Springer-Verlag, 1990.
4. Mark de Berg, Marc van Kreveld, Mark Overmars, and Otfried Schwarzkopf. *Computational Geometry. Algorithms and Applications*. Springer-Verlag, 2000.
5. G. Dissanayake, P. Newman, S. Clark, H.F. Durrant-Whyte, and M. Csorba. A solution to the simultaneous localization and map building (SLAM) problem. *IEEE Transactions of Robotics and Automation*, 2001.
6. A. Elfes. *Occupancy Grids: A Probabilistic Framework for Robot Perception and Navigation*. PhD thesis, Department of Electrical and Computer Engineering, Carnegie Mellon University, 1989.
7. Jens-Steffen Gutmann. *Robuste Navigation autonomer mobiler Systeme*. PhD thesis, University of Freiburg, 2000. (in German).
8. D. Hähnel, D. Schulz, and W. Burgard. Map building with mobile robots in populated environments. In *In Proceedings of International Conference on Intelligent Robots and Systems (IROS'02)*, 2002.
9. L. J. Latecki and R. Lakämper. Convexity rule for shape decomposition based on discrete contour evolution. *Computer Vision and Image Understanding*, 73, 1999.
10. L. J. Latecki and R. Lakämper. Shape similarity measure based on correspondence of visual parts. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 22(10), 2000.
11. Longin Jan Latecki, Rolf Lakämper, and Diedrich Wolter. Shape similarity and visual parts. In *Proceedings of the 11th International Conference on Disrecte Geometry for Computer Imagery (DGCI), Naples, Italy*, November 2003.
12. F. Lu and E. Milios. Robot pose estimation in unknown environments by matching 2D range scans. *Journal of Intelligent and Robotic Systems*, 1997.
13. T. Röfer. Using histogram correlation to create consistent laser scan maps. In *Proceedings of the IEEE International Conference on Robotics Systems (IROS-2002)*, 2002.
14. Thomas B. Sebastian, Philip N. Klein, and Benjamin B. Kimia. On aligning curves. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(1):116–125, 2003.
15. K. Siddiqi, A. Shokoufandeh, S. J. Dickinson, and S. W. Zucker. Shock graphs and shape matching. *International Journal of Computer Vision*, 35(1):13–32, 1999.
16. S. Thrun. Learning metric-topological maps for indoor mobile robot navigation. *Artificial Intelligence*, 99(1):21–71, 1998.
17. S. Thrun. Probabilistic algorithms in robotics. *AI Magazine*, 21(4):93–109, 2000.
18. S. Thrun. Robotic mapping: A survey. In G. Lakemeyer and B. Nebel, editors, *Exploring Artificial Intelligence in the New Millenium*. Morgan Kaufmann, 2002.