A longer version of the paper in IEEE Int. Conf. on Int. Robots and Systems, 2005.

# Incremental Multi-Robot Mapping

Rolf Lakaemper and Longin Jan Latecki
*CIS Dept.*
*Temple University*
*Philadelphia, PA 19122, USA*
*latecki@temple.edu, lakamper@temple.edu*

Diedrich Wolter
*Dept. of Mathematics and Informatics*
*Universität Bremen*
*Bremen, Germany*
*dwolter@informatik.uni-bremen.de*

***Abstract*— The purpose of this paper is to present a technique to create a global map of robots' surroundings by converting the raw data acquired from a scanning sensor to a compact map composed of just a few generalized polylines (polygonal curves). We propose a new approach to merging robots' maps that is composed of a local geometric process of merging similar line segments (termed Discrete Segment Evolution) with a global statistical control process.**

**In the case of single robot, we are able to incrementally build a map showing the environment the robot has traveled through by merging its polygonal map with actual scans. In the case of a robot team, we are able to identify common parts of their partial maps and if common parts are present construct a joint map of the explored environment.**

***Index Terms*— Robot Mapping, Team of Robots, Map Fusion, Cognitive Robotics, Shape Similarity**

## I. Introduction

Imagine a scenario, in which a team of robots explores an environment independently, each robot not knowing the starting pose of the others. The goal is to acquire in real time a global overview map integrating all partial maps individually collected by each robot. The partial maps are constructed from range sensor data of each robot. For example, this is a typical scenario for rescue robots, where the overview knowledge in the form of a global map is particularly important to localize victims in catastrophe scenarios (e.g., in collapsed buildings) and to ensure that the whole target region has been searched [7]. Since odometry information under such conditions is very unreliable, we assume that it is not available. We also assume that landmarks are ambiguous. According to Thrun and Liu [19] such tasks belong presently to open problems in robotics.

This paper presents an approach that makes it possible to successfully complete this task. Two main features of the proposed approach make the completion of this task possible:

- Each robot incrementally builds a map from range sensor data that is composed of a small number of generalized polylines, and exchanges its map with the other robots.
- Shape analysis and shape similarity allows determination of corresponding map parts.

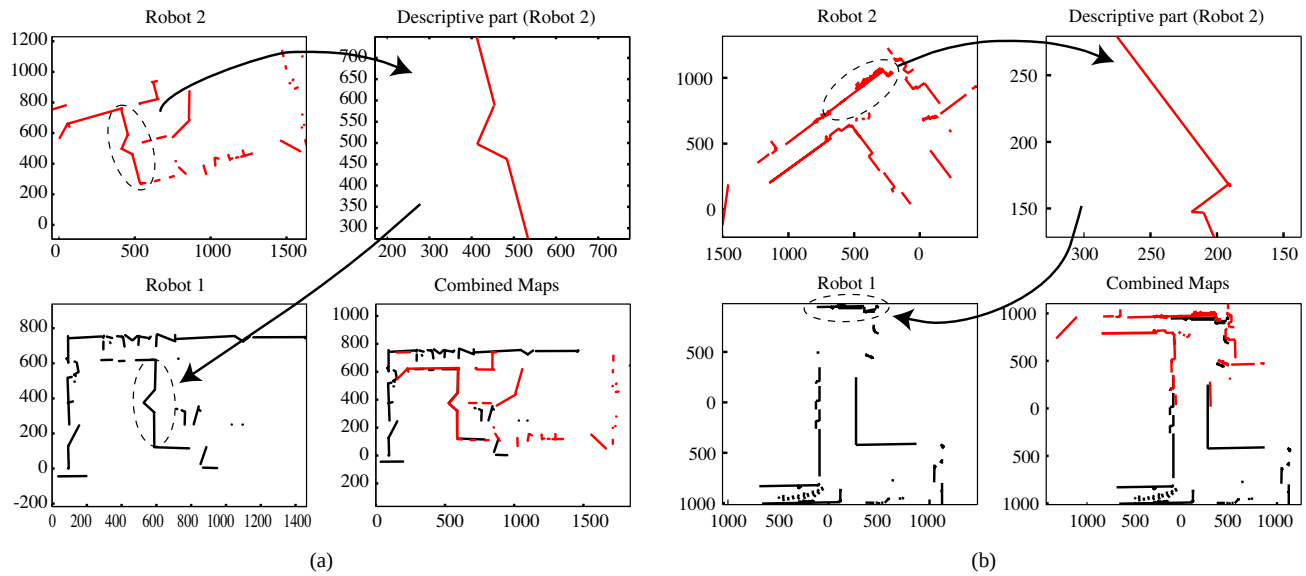
Each robot extracts parts of its polygonal map, analyzes their shape, and searches for corresponding structures of similar shapes in the maps of other robots. When two sufficiently similar structures with significant shape information are found, it is possible to align the two maps to form a joint map. The use of shape analysis adopted from high-level Computer Vision enables efficient and reliable recognition of corresponding structures as rich shape features in a local context are examined. Clearly verification is necessary in order to confirm or to reject correspondence hypotheses. The verification process involves shape similarity to compare complex polygonal structures that are close in the joint map. Aligned similar shapes confirm the map alignment hypothesis whereas dissimilar but aligned shapes contradict to it. The map fusion process can be iterated until the map alignment hypothesis is confirmed or rejected with sufficient confidence.

Two examples shown in Fig. 1 illustrate the main results of this paper obtained on real scan data. Fig. 1(a) shows two partial maps of the yellow arena from the RoboCup Rescue competition [7]. Fig. 1(b) shows two partial maps of a hallway at the University of Bremen [15]. These partial maps are reconstructed from range data by the proposed system. The proposed system was able to automatically identify a common part in both maps using shape similarity, and to align both maps to the global map shown. Finally the map merging process combines the polylines from both maps to obtain a single global map with just a few polylines.

We stress that it is shape similarity that makes possible to recognize the same structure in different maps. Once this is accomplished, a joint map of the environment can be constructed by merging the polylines of the aligned maps. We also stress that no odometry information is used in our system.

## II. Relation to Multi-Robot SLAM Approaches

There exist only a few approaches to solve the multi-robot simultaneous localization and mapping (SLAM) problem,




(a) (b)

Fig. 1. The main results of this paper obtained on real scan data: (a) yellow arena in RoboCup Rescue, (b) hallway in Bremen (unit size cm). (a) and (b) show two partial maps that are combined to form a global map based on a common part identified using shape similarity.

and their applicability depends on strong assumptions. The approach in [13] assumes that landmark identification is given using unique landmark signatures and that the initial robot poses are known. The approach in [19] enables a team of robots to build a joint map when starting poses are unknown and landmarks are ambiguous. Both statistical representation and reasoning are employed to build a joint map. The maps and robot poses are represented by Gaussian Markov random fields. Due to the complexity of the statistical reasoning, a real time implementation of this approach is questionable. Only a very simple geometric representation is used with landmarks being 2D points.

Our approach differs significantly from the approach in [19] in many aspects. Our main focus is a sophisticated geometric landmark representation. Importance of a sophisticated landmark handling in merging maps has been stressed by Konolige et al. [8].

## III. Relation to SLAM Approaches

The SLAM problem, the Simultaneous Localization and Mapping problem [3] is of high importance to mobile robotics. Consequently, it has received considerable attention (see Thrun [18] for an overview). A relation between the SLAM problem and map merging is given by the common task of place recognition, also termed the *correspondence problem*, which is part of localization. Odometry information and scan matching techniques provide good means for incremental updates to the estimated robot pose, whereas the full consequences of place recognition show up in loop closing where purely incremental pose estimation can differ significantly from the true pose. Large deviations from the true pose easily inhibit place recognition, i.e. correct localization.

Approaches to SLAM can be classified in regards to handling of uncertainty and representation of map features. The handling of uncertainty is considered the key aspect in SLAM. For this purpose 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], [6]. This means, poses are not represented as exact coordinates, but by probability distributions. To tackle loop closing within a stochastical framework, the robot's true pose must be contained in the probability distribution. Hence, these techniques provide no innate mechanism for place recognition without (good) pose estimation. This, for example, means that they fail if there is a significant discontinuity in the robot pose as is the case in the kidnapped robot problem.

A nice probabilistic framework to construct a global map from scan data is presented in [14]. However, this framework is based on the assumption that the uncertainty of scan points' positions is known. Due to the dependence of laser scan measurements on the surface characteristic of scanned objects, e.g., glass-like surface, brick wall, and metal surface, this assumption is not satisfied in our example of rescue robots. We approach the problem of constructing a global map using the principles of perceptual grouping, which look

for geometric structures in the data without any assumptions about the error characteristics [11].

Expectation maximization (EM) is another powerful technique to address the uncertainty problems. EM effectively solves the correspondence problem by iteratively calculating the most likely path of the robot. It can successfully recognize when a cycle closes, even though in order to do so it requires multiple passes over the data. This can potentially pose a problem, since there exist situations where the calculation of the probability for all possible paths would slow the algorithm beyond acceptable amounts. At present, in any case, EM approaches represent a successful solution to the correspondence problem, albeit via computationally expensive means.

We now examine representation of map features used in SLAM, focusing on their contribution to place recognition. Map features extracted from sensor data (esp. range finder data) are either the positions of special landmarks [3], simple geometric features, especially lines [12], [15], [2], or sensed reflection points are used in a direct manner [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 [4]. Stochastic models are so powerful that even linking them to a simple geometric representation like occupancy grids already yields impressive results in standard SLAM applications. To utilize grid maps in loop closing, a pose estimation is required for scan matching [8]. Scan matching of uninterpreted scans is formulated as a minimization [12], [18], [6], hence requiring a good estimate to prevent minimization getting stuck in local minima. The performance of scan matching can however be improved by considering a wider spatial context. For example polygonal lines which capture a wider spatial context than bare reflection points have been shown to allow for scan matching even when no pose estimation is available [21]. Good exploitation of geometric features within a spatial context is claimed to be the key at hand when solving the correspondence problem in map merging [8].

In the presented approach we use polygonal curves (polylines) as geometric basic entities. Shape similarity computation as used in registration is derived from the approaches proposed in [9], [10]; adaptation to matching single-scale shape information obtained through mobile robots has successfully been demonstrated [21].

## IV. Contribution to Robot Mapping

This paper addresses three main problems in robot mapping stated in Thrun [18].

1) The measurement errors are statistically dependent, since errors in control accumulate over time, and they affect the way future sensor measurements are interpreted.
2) The second complicating aspect of the robot mapping problem arises from the high dimensionality of the entities that are being mapped, which leads to serious runtime and storage problems.
3) A third and possibly the hardest problem in robot mapping is the *correspondence problem*.

The *correspondence problem* is the trouble faced when establishing correspondence between landmarks in two different maps or between landmarks sensed and landmarks in the map. We explicitly address the correspondence problem with a sophisticated shape similarity relation. In our approach, landmarks are parts of the map represented as generalized polylines with significant shape, and the relation between landmarks is shape similarity. Representing a map by just a few polygonal curves reduces the map's dimensionality. We would like to note that our approach is cognitively inspired. If a human performs map merging, identifying structures with similar shape is central. A simple illustration of this fact is how the reader evaluates the correctness of the result shown in Figure 1.

Usually landmarks are represented as 2D points with very simple geometric relations like relative distances and angles of triples of adjacent landmarks within a small radius [19], [16]. The fact that our landmarks have significant structure, which we compare, using shape similarity, greatly reduces the possibility of landmark confusion. We want to stress that the enhanced geometric representation of landmarks is independent from, and is compatible with other techniques of landmark identification.

We address the problem of measurement errors being statistically dependent with a new process of map merging that is based on geometric local process of line segment merging with a global statistical control.

In our map representation, we simply do not run into the problem of high-dimensionality, since our representation is built of higher level objects, which are line segments and generalized polylines. We have an explicit process, called Discrete Segment Evolution, that reduces the number of line segments to a minimal number required to represent the mapped environment.

## V. Single Robot Mapping

In this section we introduce some notation regarding the system used by a single robot to create its global map of its surroundings, and we summarize the main steps performed at each iteration of the algorithm, i.e., on the arrival of a new scan. Videos illustrating the incremental mapping results can be viewed on http://knight.cis.temple.edu/∼shape/robot/.

A global map from the view of a single robot is termed a partial global map from the view of a robot team.

A global map of a single robot is built iteratively as the robot moves. We denote the scan and the (partial) global map at time $t$ by $S_t$ and $G_t$, respectively. Global map $G_t$ and scan $S_t$ is composed of *generalized polylines*.

A *generalized polyline* is a set of line segments, having a specific ordering, whose vertices may or may not be connected. Observe that a classical definition of a polyline (polygonal curve) requires that the endpoints of consecutive segments coincide. Generalized polylines result naturally when scan points are approximated with line segments, which is our first processing step of the input range data. By dropping the constraint that a polyline be composed of line segments whose vertices are connected, we can better approximate the scan points, especially if obstacles with curved boundaries are sensed. The usage of generalized polylines is particularly important in the polyline merging algorithm described below.

Our first processing step (approximation of scan points with line segments) is followed by the segment grouping step. We form an ordered list of segments by minimizing the sum of the distances of their endpoints. Finally, if the endpoints of consecutive segments are too far apart, we split the list into sublists. Thus, generalized polylines are sublists of this list.

To create a (partial) global map $G$, we start with the first global map $G_1$ being equal to the first scan $S_1$. Henceforth, assuming we have created the global map $G_{t-1}$ at time $t-1$ and a new scan $S_t$ has arrived, $G_t$ is created in the following three steps:

- **Correspondence** Using the structural shape similarity [21] that is briefly described in Section VIII-B, a set of corresponding polylines is chosen between the current scan $S_t$ and the global map $G_{t-1}$. The selected polylines may be parts of the polylines in $S_t$ and $G_{t-1}$, since new objects appear in $S_t$ and the existing objects in $G_{t-1}$ may partially disappear in $S_t$. We denote the corresponding polylines $PL_i^S$ and $PL_i^G$, which belong to sets $S_t$ and $G_{t-1}$ respectively, with the same index $i$, for $i = 1, ..., n$.
  The shape similarity measure defined in [21] does not depend on the position of the polylines in the map. Therefore, it also works in the presence of discontinuities in the robots motion (e.g., kidnapped robot) and allows to detect closing of a loop. Observe that existing map matching approaches are based on closest points, e.g., [6], and therefore, depend on the position of the objects in the maps.
- **Alignment** The current scan $S_t$ is rotated and translated until a minimum distance is found between the points in corresponding polylines $PL_i^S$ and $PL_i^G$. We align scan $S_t$ and map by extending the Iterative Closest Point (ICP) algorithm [5] to respect the correspondence of polylines.
- **Merging** (A detailed discussion follows in Section VI.) The output of alignment overlays the actual scan on the global map, but the surfaces of the same objects are still represented by separate polylines. The goal of merging is to represent surfaces of the same objects by single polylines.

Results obtained by this algorithm are illustrated in Figure 2. The Figure shows to the left input data which presents lines extracted from 400 laser scanner measurements. The Figure's right hand side depicts the resulting fused map which only contains few polygonal lines resembling the map information.

## VI. Merging

Once $S_t$ has been aligned to $G_{t-1}$, merging consists of two steps which integrate the new information contained in $S_t$ with the previous global map to produce $G_t$. In other words, this module takes as input the new scan, $S_t$, the previous global map, $G_{t-1}$, and the sets of corresponding polylines $PL_i^S$ and $PL_i^G$ and outputs $G_t$, the new and current global map. The two steps of this process are pair creation and simplification.

- **Pair creation** We create all possible pairs of line segments that are sufficiently similar, taking one segment from $S_t$ and $G_{t-1}$ each. The similarity of line segments is measured with the cost function $C$ described below. We replace both line segments with a new line segment.
- **Simplification** considers all line segments, including the newly created ones, without regard to whether they belong to $S_t$ or $G_{t-1}$. Pairs of line segments are merged together to form new segments using the same cost function $C$. However, a simplification process has a second global control mechanism: We iteratively merge a pair whose merging cost $C$ is the lowest at each pass. This process can be implemented in $N \log(N)$ time, where $N$ is the number of line segments, since it is enough to find the closest single partner for each line segment and sorting the values of $C$ requires $N \log(N)$ time.

We use the histogram of the line segment directions to find main directions that the newly created line segments must follow. This statistical control provides a solution to the problem of cumulative errors. Accumulative errors introduce systematic distortions in the directions of line segments that accumulate slowly. The main issue here is that these accumulative errors do not lead to peaks in the direction

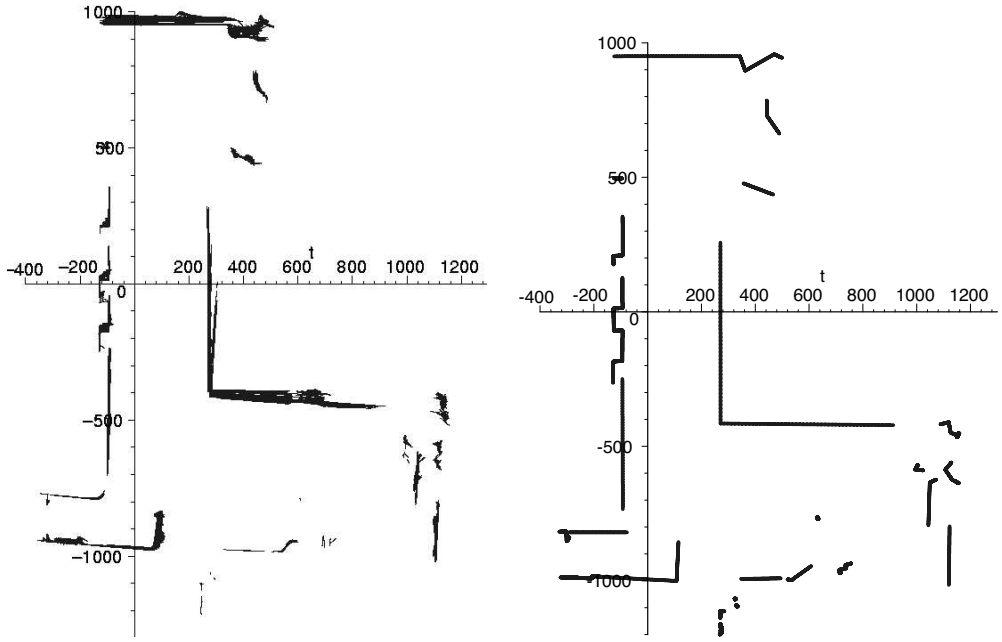



Fig. 2. The right figure is a simplification of the left map containing lines from 400 aligned laser range finder scans. The Figure's unit size is cm; the data has been collected by a robot traversing a hallway at Bremen university.

histogram, and consequently, appearing line segments are correctly mapped to the existing main directions. On the other hand, if a surface of a new object is oriented into a new direction, it will lead to a peak in the direction histogram after a few scans of the surface have been acquired. The fact that direction histogram provides a solution to the problem of cumulative errors is also true for other histogram-based approaches, in particular for [15].

### *A. Main directions*

The main directions are obtained as significant peaks in the direction histogram of line segments (angles with the x axis). Each line segment in the global map plus the aligned new scan $G_t^p = G_{t-1} \cup S_t$ contributes to the bin representing its direction with the weight that is its length.

## VII. Line segment merging based on perceptual grouping

Given a pair of line segments, $L_1$ and $L_2$, and the angular direction $ad$ the objective of the merging process is to compute a merged segment $ms(L_1, L_2, ad)$ with the angular direction $ad$. The main idea is to only merge line segments that are sufficiently congruent. Therefore, we need to define a cost function $C(L_1, L_2, ad)$ that measures the congruency of $L_1$ and $L_2$ in the context of the main direction $ad$. The main direction $ad$ is either defined by the histogram of directions (Section VI-A) or is computed as an average direction of $L_1$ and $L_2$ weighted with the lengths of $L_1$ and $L_2$. Let $L_1 = AB$ and $L_2 = CD$ be oriented in the same direction so that $||AB - CD|| \leq ||AB + CD||$ (see in Fig. 3), i.e., the scalar product $< AB, CD > \geq 0$, then the weighted average direction of $L_1$ and $L_2$ is obtained as the direction of vector $AB + CD$. Although we use the weighted average direction in the computation of the proposed cost function, the actual merging direction $ad$ is obtained from the histogram of directions of all line segments in the given map (in which case merging would also have a direction quantization effect that is needed to cope with accumulative errors).

The geometric intuition of the presented merging process and, in particular, the definition of merging cost $C(L_1, L_2, ad)$ is based on cognitively motivated principles of perceptual grouping. In particular, we followed the approach presented in [11] on grouping line segments to form longer line segments. It states that proximity of endpoints, parallelism, and collinearity are the main geometric relations that influence the perceptual grouping of line segments. Our setting is slightly different, since we merge two line segments only with respect to a given main direction $ad$. Therefore, we developed a new cost function that integrates these geometric relations. For illustration see Fig. 3, where $L_1 = AB$ and $L_2 = CD$.

- **Parallelism:** The less parallel $L_1$ and $L_2$ are to the target direction $ad$, the greater the cost of merging $L_1$ and $L_2$

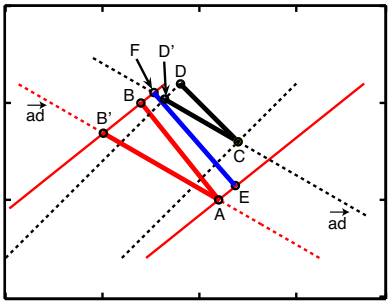



Fig. 3. Illustration to merging cost computation. Lines $AB$ and $CD$ are merged in the context of main direction $ad$; $EF$ shows the average direction.

to a segment following direction $ad$. To measure the parallelism between $L_1$ and direction $ad$, we position a line with direction $ad$ through one of the endpoints of $L_1$, point $A$ in Fig. 3. Then we find the intersection point $B'$ between the line orthogonal to $L_1$ through $B$ and the direction line. In a similar way, we find the intersection point $D'$ for segment $L_2$. The parallelism measure is defined as

$$parC(L_1, L_2, ad) = \frac{l_1}{l_1 + l_2} d(B, B') + \frac{l_2}{l_1 + l_2} d(D, D'),$$

where $d$ is the Euclidean distance, and $l_1, l_2$ are the lengths of segment $L_1, L_2$, correspondingly. The weight factors are so that the longer segment has larger influence on the cost value. If the direction $ad$ is orthogonal to one of the segments, then $parC(L_1, L_2, ad) = \infty$. Observe that $parC(L_1, L_2, ad) = 0$ iff $L_1, L_2, ad$ are all parallel.

- **Collinearity:** The intuition is again clear. The closer are two segments $L_1, L_2$ to being collinear, the smaller should their collinearity measure be. Let $lwa$ be a straight line with a direction that is an average of $L_1$ and $L_2$ directions weighted with lengths of $L_1$ and $L_2$. It is positioned between $L_1$ and $L_2$ in a way described below. $lwa$ is defined by segment $EF$ in Figure 3. The greater the distance of segments $L_1$ and $L_2$ from $lwa$, the less collinear they are, and consequently, the higher the value of the cost function. The collinearity measure is defined as the sum of maximal distances between the line $lwa$ and segments $L_1$ and $L_2$:

$$colC(L_1, L_2) = \max\{d(A, lwa), d(B, lwa)\} + \max\{d(C, lwa), d(D, lwa)\},$$

where $d(X, lwa)$ is the distance between point $X$ and line $lwa$. Observe also that $colC(L_1, L_2) = 0$ iff $L_1$ and $L_2$ are collinear.

- **Proximity:** The greater the distance between segments $L_1$ and $L_2$, the higher the value of the cost function. The proximity cost is defined as

$$proxC(L_1, L_2) = \min\{d(A, CD), d(B, CD), d(C, AB), d(D, AB)\},$$

where $d(X, YZ)$ is the distance between point $X$ and line segment $YZ$. Observe that $proxC(L_1, L_2) = 0$ iff one of the endpoints of one segment lies on the other segment.

Finally, the cost of merging segments $L_1$ and $L_2$ to a segment following the main direction $ad$ is defined by:

$$C(L_1, L_2, ad) = w_1 \cdot parC(L_1, L_2, ad) + w_2 \cdot colC(L_1, L_2) + w_3 \cdot proxC(L_1, L_2),$$

The weights are used to obtain an adequate balance between parallelism, collinearity and proximity of the line segments. In our approach they were determined experimentally and set to $w_1 = 2, w_2 = \frac{1}{4}, w_3 = \frac{1}{2}$.

Now we describe how a line $ld$ with a given direction $ad$ is positioned between segments $L_1$ and $L_2$. The direction $ad$ can be a given main directions or can be a weighted average direction of segments $L_1$ and $L_2$ as it is the case for line $lwa$ in the definition of the collinearity cost.

The line $ld$ with direction $ad$ is positioned between $L_1$ and $L_2$ so that the following equation

$$d_1 \cdot l_1 = d_2 \cdot l_2$$

is satisfied, where $l_i$ is the length of segment $L_i$ and $d_i$ is the distance of the midpoint of $L_i$ to line $ld$ for $i = 1, 2$. This has the effect of positioning line $ld$ closer to the larger of two segments.

Finally, for a given main direction $ad$, the segment obtained by merging $L_1$ and $L_2$, called the merged segment $ms(L_1, L_2, ad)$, is defined by the convex hull of the projections of $L_1$ and $L_2$ on line $ld$.

## VIII. Map Fusion

Map fusion, the task of merging partially overlapping maps being built by different robots, mainly consists of aligning these maps and therefore finding corresponding areas/objects first. For the correspondence a shape similarity measure that is position independent is necessary to identify common structures in both maps. Since there is no information on the expected robot position across the different maps any position dependent measure of polylines, e.g., Hausdorff distance, can not be applied. Once the correspondences are found and the maps are aligned, the merging process itself is similar to the single robot approach (Section V).

We assume that the input polygonal maps are as described in V. However, the input polygonal maps can also be obtained from different robot mapping systems (e.g., [21]) or by post-processing grid maps (e.g., [20]). Merging polygonal maps is achieved by congruently aligning them, i.e. transforming the maps into a single coordinate system, and merging corresponding objects, i.e. corresponding polylines. Note that alignment is implied by correspondence of polylines. The key challenge in map merging is to solve the correspondence problem with an approach that promises a real time implementation. Theoretically, any pair of objects from different maps may correspond. This renders any naive approach to search for the most likely correspondence of objects infeasible since combinatorial explosion occurs.

We propose the use of position independent shape analysis and shape similarity measures to tackle the correspondence problem. Following a cognitive inspiration, we iteratively merge maps, by repeatedly determining the most *salient* object and finding a hypothetic, *matching* counterpart to refine the maps' alignment. The most plausible correspondence relation is used (if it exists) to merge individual polylines into a single, coherent map.

The algorithm basically describes a guided search in the space of two maps' possible correspondences. It is based on the heuristic that the probability of wrong correspondence decreases with the shape complexity of corresponding polygonal surfaces. The algorithm to determine the most plausible correspondence relation proceeds as follows (for the ease of description we assume that only two maps are to be merged).

**Input:** Two maps $M_1$, $M_2$ consisting of polylines $\{P_1, \ldots, P_n\}$ and $\{Q_1, \ldots, Q_m\}$ respectively.

**Hypothesis selection:**

H1 Select the most salient object $P_i$ of map $M_1$ as sequence of consecutive line segments with maximal shape complexity $sc$ (defined below) that has not been selected before in hypothesis selection. Observe that $P_i$ is a generalized polyline. If the selected $P_i$ does not have sufficient shape complexity ($sc(P_i)$ is not above a predefined threshold), terminate the algorithm with the decision that the maps cannot be fused.

H2 Using a shape similarity measure $s$ (defined below) find a polyline $Q_j$ of map $M_2$ for which $s(P_i, Q_j)$ is minimal, i.e. $P_i$ and $Q_j$ show off maximum similarity. The correspondence hypothesis $P_i \sim Q_j$ is then checked in the validation step.

**Validation:**

V1 Based on the correspondence $P_i \sim Q_j$ compute an alignment of $M_1$ and $M_2$ that congruently aligns $P_i$ and $Q_j$. This is based on shape similarity $s(P_i, Q_j)$ that not only measures shape differences, but also determines the points' correspondence and the rotation needed to align the polylines $P_i$ and $Q_j$.

V2 Using the alignment parameters (angle and translation) from (V1), we fuse both maps $M_1$ and $M_2$ to a common coordinate system. Now we can apply a position dependent similarity measure to verify the correctness of the alignment. We identify two parts $M_1'$ of $M_1$ and $M_2'$ of $M_2$ that are close with respect to Hausdorff distance $H$, i.e., $H(M_1', M_2')$ is below a predefined threshold. If among the polylines of $M_1'$ we can find the at least one polyline with sufficient shape complexity, the fusion hypothesis is confirmed. Otherwise it is rejected, and we go back to (H1).

Now we describe the algorithm's building parts, namely shape complexity and shape similarity.

### A. Shape Similarity

We utilize a position independent shape similarity measure $s$ to compare the shape of polygonal curves. Technically speaking, a shape similarity measure is a shape distance measure as maximal similarity of comparing identical shapes yields $0$, the lowest value possible. This might be a bid confusing but follows the definitions of shape similarity in Computer Vision (e.g. [1]). The following definition applies to classical as well as to generalized polylines.

Let a polygonal curve $P = \{p_1, ..., p_n\}$ be given as a sequence of line segments. A tangent space or (turn angle) representation $T(P)$ is defined as step function that assigns the angle with $x$ axis to each line segment $p_i$ and the length of each step is equal to the length of the original line segment.

The dissimilarity measure of two polylines $C$ ad $D$ with normalized arc length is defined as

$$s(C, D) = \int_0^1 (T(C)(t) - T(D)(t) + \Theta(C, D))^2 \; dt$$

where $\Theta(C, D)$ is chosen to minimize the integral (it accounts for different orientation of curves) and is given by $\Theta(C, D) = \int_0^1 (T(C)(s)T(D)(s))^2 \, ds$ (see [1] for details). $\Theta(C, D)$ gets reused as our first estimate of the rotation angle needed to align both polylines. An improvement to handling of very noise data lacking of rich shape information and to handling of very complex shapes is presented in [10], [9]. Experimental results for shape similarity measures applied to range finder data is presented in [21].

### B. Shape Complexity

The definition of shape complexity applies to generalized as well as classical polylines. Given a polyline $P$ we again consider the tangent space representation as introduced in the previous section. Shape complexity has been defined

in accordance to the presented shape similarity measure. The underlying idea of shape complexity as used in the experiments here is to define complexity as deviation from the most simple polygonal curve, a straight line. A natural way to measure difference in our context is to use the shape similarity measure. Shape complexity of a polyline $P$ is defined[1] as $sc(P) = s(P, L_P)$ where $L_P$ denotes a straight line with same curve length as $P$.

### C. Implementation Details

Finding a shape similar polyline in $M_2$ (HS2) for a given query salient polyline $P_i$ from $M_1$ is implemented using a sliding window on the list of line segments in $M_2$. Thus, the complexity is linear with respect to the number of line segments. This number is extremely low (on the order of 30 to 50 segments), since we work with simplified maps.

The verification process (V2) is also linear with respect to the number of line segments, since Hausdorff distance of line segments can be implemented by considering their endpoints.

Thus, the complexity of the map fusion depends mainly on the number of iterations of hypothesis selection and verification steps. In all our experiments just a few iterations were sufficient. Clearly, the number of iterations is bounded by the number of parts with sufficiently salient shape.

## IX. Conclusions

The paper describes a novel method to create a global map of robot surrounding by converting the data to a compact map composed of a small number of generalized polylines. This method can be used not only in creating a global map with many scans obtained throughout the scanning process of one moving robot, but also partial maps from many different robots. The process is performed without using odometric information.

The main contribution of the paper is fusing of partial maps to a single map of the environment. One of the main problems in map fusing is the correspondence problem (i.e., which part of one map corresponds to which part of the other maps). The correspondence problem is solved by using shape similarity measure that is based on position independent matching between polylines with high shape complexity.

## Acknowledgment

This work was supported in part by the National Science Foundation under grant INT-0331786 and grant R3 [Q-Shape] in the framework of the SFB/TR 8 Spatial Cognition from German Research Foundation (DFG). We would like to thank Sebastian Thrun for a helpful discussion regarding the topic of this paper. Thomas Röfer and Adam Jacoff are acknowledged for providing scan data.

[1]This is an alternate definition of shape complexity to the measure defined in [21]. Results are, however, similar.

## 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 Trans. PAMI*, 13:209–206, 1991.

[2] 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.

[3] 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.

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

[5] A. Fitzgibbon. Robust registration of 2d and 3d point sets. In *Proc. British Machine Vision Conference, volume II, Manchester, UK*, pages 411–420, 2001.

[6] D. Hähnel, D. Schulz, and W. Burgard. Map building with mobile robots in populated environments. In *Proceedings of International Conference on Intelligent Robots and Systems (IROS'02)*, 2002.

[7] A. Jacoff, E. Messina, and J. Evans. Performance evaluation of autonomous mobile robots. *Industrial Robot: An Int. J.*, 29(3), 2002.

[8] K. Konolige, D. Fox, B. Limketkai, J. Ko, , and B. Stewart. Map merging for distributed robot navigation. In *Intl. Conf. on Intelligent Robots and Systems*, pages 212–217, 2003.

[9] 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.

[10] L. J. Latecki, R. Lakämper, and D. Wolter. Partial optimal shape similarity. *Image and Vision Computing J.*, 23(2):227 – 236, 2005.

[11] D. G. Lowe. Three-dimensional object recognition from single two-dimensional images. *Artificial Intelligence*, 31:355–395, 1987.

[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] E. Nettleton, H. Durrant-Whyte, P. Gibbens, and A. Goktogan. Multiple platform localization and map building. In G. T. McKee and P. S. Schenker, editors, *Sensor Fusion and Decentralized Control in Robotic Systems III*, pages 337–347. Bellingham, 2000.

[14] S. T. Pfister, S. I. Roumeliotis, and J. W. Burdick. Weighted line fitting algorithms for mobile robot map building and efficient data representation. In *ICRA*, 2003.

[15] T. Röfer. Using histogram correlation to create consistent laser scan maps. In *Proceedings of the IEEE International Conference on Robotics Systems (IROS-2002)*, 2002.

[16] J. D. Tardos, J. Neira, P. M. Newman, and J. J. Leonard. Robust mapping and localization in indoor environments using sonar data. *Int. J. Robotics Research*, (5(4)):311–330, 2002.

[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.

[19] S. Thrun and Y. Liu. Multi-robot slam with sparse extended information filters. In *Proc. of ISRR, Sienna, Italy*, pages 322–331. ACM, 2003.

[20] M. Veeck and W. Burgard. Learning polyline maps from range scan data acquired with mobile robots. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2004.

[21] D. Wolter and L. J. Latecki. Shape matching for robot mapping. In C. Zhang, H. W. Guesgen, and W. K. Yeap, editors, *Proc. of 8th Pacific Rim Int. Conf. on Artificial Intelligence*, Auckland, New Zealand, August 2004.