Building Polygonal Maps from Laser Range Data

Longin Jan Latecki [1] and **Rolf Lakaemper** [2] and **Xinyu Sun** [3] and **Diedrich Wolter**[4]

Abstract. This paper presents a new approach to the problem of building a global map from laser range data, utilizing shape based object recognition techniques originally developed for tasks in computer vision. In contrast to classical approaches, the perceived environment is represented by polygonal curves (polylines), possibly containing rich shape information yet consisting of a relatively small number of vertices. The main task, besides segmentation of the raw scan point data into polylines and denoising, is to find corresponding environmental features in consecutive scans to merge the polyline-data to a global map. The correspondence problem is solved using shape similarity between the polylines. The approach does not require any odometry data and is robust to discontinuities in robot position, e.g., when the robot slips. Since higher order objects in the form of polylines and their shape similarity are present in our approach, it provides a link between the necessary low-level and the desired high-level information in robot navigation. The presented integration of spatial arrangement information, illustrates the fact that high level spatial information can be easily integrated in our framework.

1 INTRODUCTION

The problems of self-localization and robot mapping are of high importance to the field of mobile robotics. Robot mapping describes the process of acquiring spatial models of physical environments through mobile robots. Self-localization is the method of determining the robot's position with the robot's internal spatial representation. The central method required is a matching of sensor data, which - in the typical case of a laser range finder as the robot's sensor - is called scan matching. Whenever a robot needs to cope with unknown or changing environments, localization and mapping have to be carried out simultaneously, this technique is called SLAM (Simultaneous Localization and Mapping). To attack the problem of mapping and/or localization, mainly statistical techniques are used (Thrun [15], Dissanayake et al. [3]), e.g., the extended Kalman filter, a linear recursive estimator for systems described by non-linear process models and/or observation models, are the basis for most current SLAM algorithms. Bayesian rules build the foundation of the models employed. For localization, often partially observable Markov decision processes (POMDP) are utilized.

The robot's internal geometric representation forms the basis for these techniques. It is build atop of the perceptual data read from the laser range finder (LRF). Typically, either the planar location of reflection points read from the LRF is used directly as the geometric representation, or simple features in the form of line segments or corner points are extracted (Cox [2]; Gutmann and Schlegel [5]; Gutmann [7]; Röfer [14]). Although robot mapping and localization techniques are very sophisticated they do not yield the desired performance. We observe that these systems use only a very primitive geometric representation. As the internal geometric representation is a foundation for the sophisticated techniques in localization and mapping, shortcomings on the level of the geometric representation affect the overall performance. The main goal of this paper is the introduction of an elaborate and cognitively motivated geometric representation and a reasoning formalism for robot mapping. A successful geometric representation must result in a much more compact representation than uninterrupted perceptual data, but must neither discard valuable information nor imply any loss of generality. We claim that the representation proposed in this paper, namely polygonal curves or polylines, representing parts of object surfaces being obtained from segmented scans, fulfills these demands. The relation among the objects is based on shape similarity and on qualitative arrangement information. Representing the passable space explicitly by means of shape is not only adequate for mapping applications but also helps to bridge the gap from metric information needed to topological knowledge due to the object centered perspective offered. Moreover, an object-centered representation is a crucial building block in dealing with changing environments, as such a representation allows us to separate the partial changes from the unchanged parts. In this paper we focus on incremental building of the object representation.

There exist approaches to map building that apply more sophisticated geometric method to scan data, e.g., Forsberg et al. [4] and Jensfelt and Christensen [9]. However, they focus on extraction of linear structures only, why we not only consider extraction of polygonal structures but also on similarity of polygonal structures. The similarity of polygonal structures is a driving force of our approach.

2 MAP BUILDING PROCESS

Map Building is the process of memorizing perceived objects and features the robot has passed by, merging corresponding objects in consecutive scans of the local environment. The robot's internal spatial representation is referred to as a map, in the case of a feature based spatial representation it is commonly referred to as a feature map [15]. A key challenge in map building is to match a local sensor reading against the global map. Multiple problems occur, e.g., the noise of the data perceived must be filtered in a way to obtain the required features, and the correspondence between perceived objects must be found on the basis of the filtered (visual) features, since additional information, i.e., odometry, has proven to be inaccurate. This excludes the possibility of simply superimposing consecutive scans on top of one another, as can be seen in the example shown in Figure 1(a). It shows the effects of accumulated errors in rotation and distance measure, because the geometrical properties of the envi-

[1] Temple University, Philadelphia, USA, email: latecki@temple.edu
[2] email: lakamper@temple.edu
[3] email: xysun@euclid.math.temple.edu
[4] University of Bremen, Bremen, Germany email: dwolter@informatik.uni-bremen.de

ronment indicated by the LRF purely interpreted in connection with odometry increasingly differ from reality, resulting in large displacements of the perceived objects and block-like representations in the global map. Odometry is designed to record the distance the robot has traveled and the rotational angle the robot has turned with respect to the starting position. However, odometry makes unreliable recording on a large scale.

Using alignment based on shape similarity, we can construct a significantly better map by correcting the errors of the odometry or even discarding the recording of the odometry completely. The alignment is computed using stable objects in the map to calculate the current position of the robot. Assuming that some objects in two consecutive scans are not moving, we align the stationary objects in the second scan to the corresponding objects in the first scan. The robot's movement and its position defining the global position of the scanned environment is achieved by the resulting movement and rotational angle computed by the alignment. An example global map computed using our alignment algorithm is shown in Figure 1(c). The scans are aligned iteratively, then superimposed on each other without any information from the odometry. The row scan data alignment based on robot's odometry is shown in Figure 1(a). The quality of the map in (c) is much better, and the objects can be easily identified. To have a fair comparison, the map in (c) should be compared to the map in (b). The map (b) is obtained from (a) with a simple algorithm that corrects significant rotation errors of the odometry by finding rotation angles that maximize the proximity of long lines and rotated according the scans accordingly. Although the improvement from (b) to (c) is significant, we can still find some fuzziness in certain areas of (c). This is due to the nature of the perceived data, which introduces noise in several ways:

1. Scanning an object with a fractal or nonrigid shape. An example of such can be a plant in any office building. The locations of the scan points on such objects are mostly random, and the recordings of the scan points are not going to provide us useful position information, even if the robot is *not* moving.
2. Scanning objects too far away. Scanning objects far away inevitably will create more error than scanning objects close by, due to uneven ground or other movement/vibration effects. Aligning all objects in the scans, the error introduced by this defective data will accumulate and propagate.
3. Scanning moving objects. Objects are aligned under the assumption that only the robot moves, being the only source of change of the distance between them. But once there is a moving object in the scan, such an assumption is no longer true, and the alignment will not be reliable, especially when the moving object is near the scanner.

In the following we will address these problems, and discuss their solutions in our framework.

One shortcoming of the recursive alignment process is that when errors occur, they will not be corrected nor eliminated. In fact, they are most likely to be accumulated and propagated as show in Figure 1(c). Another drawback is that each scan still exists independently of the others, i.e., we are not creating *a global map* composed of objects in the common sense of the word, but composed of a set of unrelated polylines. Hundreds of laser scans are printed on paper and human eyes can easily identify objects from the printout, but these scans cannot be used by robot localization because these objects are not present in the internal robot representation.

Hence a process is needed that can deal with these problems, and that can create a global map composed of just a few objects. We propose such a process that we call merging in this paper. In order to describe it, we need to introduce some notation.

A global map is built iteratively as the robot moves. We denote the scan and the global map at time t by S_t and G_t respectively. Each scan S_t and the global map G_t is composed of polylines. We assume that the range data is mapped to locations of reflection points in the Euclidean plane, using a local coordinate system. These points are segmented into individual polylines with a simple heuristic: Traversing the reflection points following the order of the LRF, an object transition is said to be present wherever two consecutive points are further apart than a given distance threshold (20 cm in the case of our example map). The precise choice of the threshold is not crucial as subsequent processing accounts for differences. The obtained polygonal objects are further simplified to reduce the influence of noise (Section 3.1). We apply discrete curve evolution (DCE, described below) followed by least square fitting of line segments to obtain the simplified polyline objects.

To create the global map G, we start with the first global map G_1 being equal to the first scan S_1. Assuming we have created the global map G_{t-1} at time $t-1$, we use G_{t-1} and S_t to create G_t in the following steps:

- We extract a virtual scan $\tilde{S}_{t-1}$ from the global map G_{t-1}. $\tilde{S}_{t-1}$ is the part of the global map to which the previous scan S_{t-1} was merged. The virtual scan $\tilde{S}_{t-1}$ is a corrected version of the actual scan S_{t-1}. The noise correction was achieved by merging in the previous step of S_{t-1} to the global map G_{t-2}.
- We use shape similarity (described in Section 3) to find the correspondence between objects in S_t and $\tilde{S}_{t-1}$, which can be a many-to-many correspondence. Since $\tilde{S}_{t-1}$ is part of the global map G_{t-1}, we obtain the correspondence between S_t and parts of the global map.
- We align polylines in S_t to the corresponding polylines in G_{t-1}. We repeatedly apply the least squares method to find the optimal translation and rotation. The main difference in comparison to the standard approaches is that the corresponding points are limited to polylines with similar shape. Thus, we greatly reduce the problem of local minimum. This allows us to align S_t to G_{t-1} even if the robot displacment is large (see Figure 3).
- Finally, we merge the aligned scan S_t to G_{t-1} to create G_t. Merging an aligned new scan to the global map adds newly detected polylines in the surrounding area to the global map while not discarding the existing polylines no longer seen by the robot. The goal is to produce a global map composed of a few polylines. The details of merging are presented in Section 4.

Repeating this procedure iteratively for each new scan, we are able to create a global map that remembers all the objects and features along the path the robot has traveled. The result can be seen in Figure 1(d), which displays the final global map as a set of polylines. We used exactly the same raw input data for all four maps in Figure 1. This feature-based approach yields a compact representation, and most importantly, it represents the robot's surroundings by a single set of non-overlapping polylines that can be easily recognized and compared using shape similarity.

3 STRUCTURAL REPRESENTATION OF SHAPE

A first step in the presented approach is to extract shape information from data acquired by a laser range finder (LRF). Polygonal lines, termed polylines, serve in our approach as a fundamental building

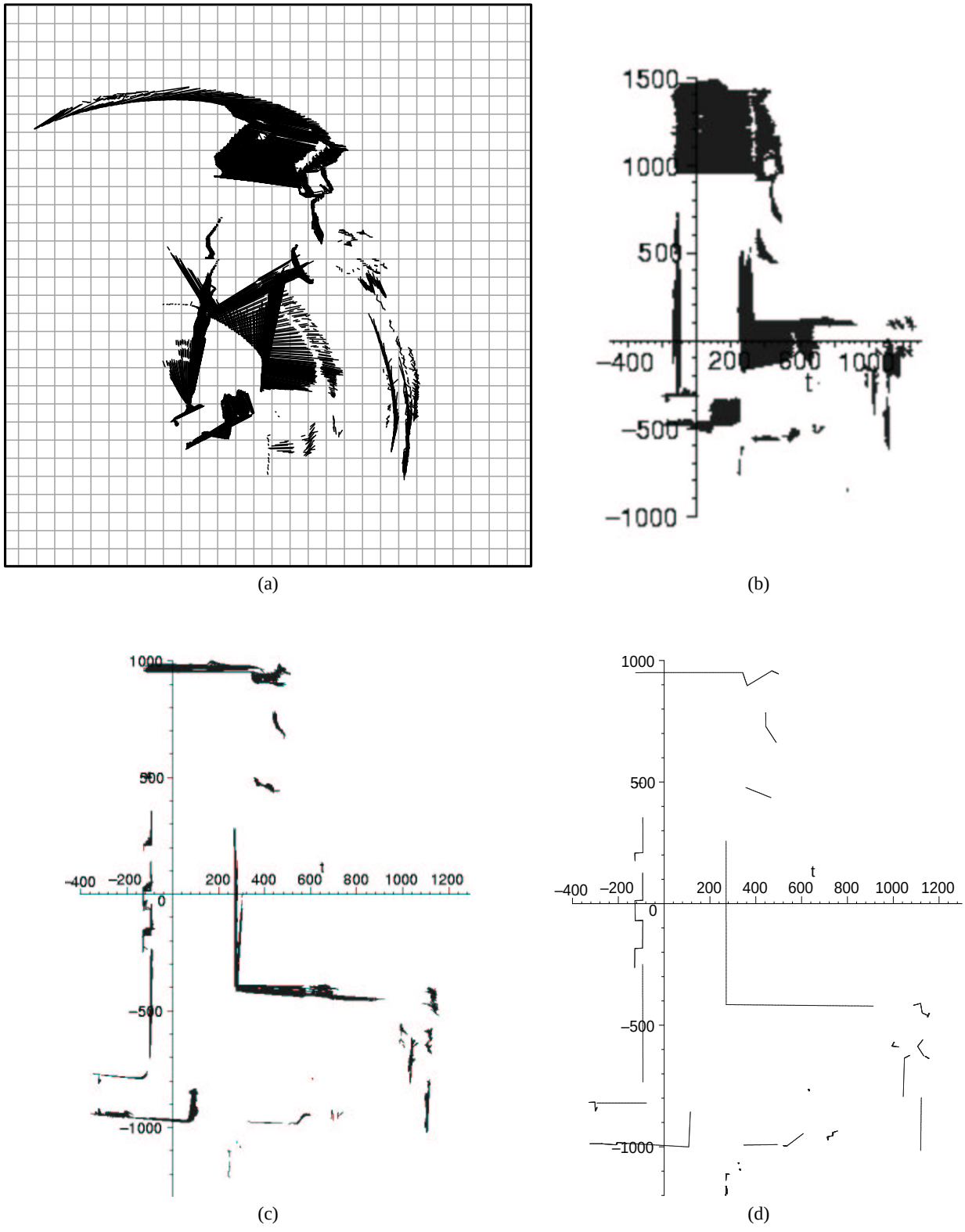

Figure 1. (a) A global map built using the information from odometry only. (b) A global map with significant rotation errors corrected. (c) A global map built using the proposed alignment only. (d) A real global map built using the proposed merging approach.

block. They already capture more context than other features typically employed in scan matching approaches (e.g., simple line segments or even uninterpreted data). The richness of perceivable shapes in a regular indoor scenario yields a more reliable matching than other feature-based approaches, as mixups in determining 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.

3.1 Grouping and Simplification of Polylines

Polylines extracted from raw scan data 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 [11, 12] which achieves these goals without losing valuable shape information. DCE is a context-sensitive process that proceeds iteratively. 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[5]:

$$K(u, v, w) = |d(u, v) + d(v, w) - d(u, w)|$$

Hereby, d denotes the Euclidean distance. The process of DCE is very simple and proceeds in a straightforward manner. The least relevant vertex is removed until this relevance measure exceeds a given threshold, thereby defining the level of polygon simplification. Consequently, as no relevance measure is assigned to end-points, they remain fixed. The choice of a specific simplification threshold is not crucial, refer to Figure 2 for results. Implementation of DCE can benefit from the observation that a polyline can be represented simultaneously as a double-linked list and a self-balancing tree which reflects the order of relevance measures. Thus, the overall complexity is $O(n \log n)$. Proceeding this way we obtain a cyclic, ordered vector of polylines.

3.2 Similarity of Polylines

Matching scans against the global map within the context of a shape based representation is naturally based on shape matching. This somehow revives a notion in Lu and Milos' fundamental work [13] *"scan matching is similar to model-based shape matching"* that so far has not received much attention. In the presented approach we adopt a shape matching originated from computer vision that has proven successful in the context of shape retrieval [12]. It may easily be adapted to our needs. The property of invariance to change of scale as often desired in computer vision approaches is not adequate in our domain and must be excluded.

To compute the similarity measure between two polygonal curves, we establish the best possible correspondence of maximal convex arcs, where a *convex arc* is a left- or right-arcuated arc. To achieve this, we first decompose the polygonal curves into maximal subarcs that 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 convex arc that corresponds to a part of the other contour composed of adjacent maximal arcs. The best correspondence, i.e., the one yielding the lowest similarity measure, can be computed using dynamic programming, where the similarity of the corresponding visual parts is as defined below. Using dynamic programming, the similarity between corresponding parts is computed and aggregated. The computation is described extensively in [12]. 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 of line-segments only. This representation was previously used in computer vision, in particular in [1]. Denoting the mapping function by T, the similarity gets defined as follows:

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

where $l(C)$ denotes the arc length of C, and the whole integral is over arc length. The constant $\Theta_{C,D}$ is chosen to minimize the integral (cp. [12]). 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 (cp. [12]). 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; position and orientation will be accounted for when computing the actual matching.

3.3 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. We must also take into account that (a) not all polylines may get matched as the features' visibility changes and (b) that due to segmentation noise (cp. section 3.1) it is not necessarily a one-to-one correspondence. Furthermore, any correspondence of polylines induces an alignment of the polylines which constraints the scan's origin. For example, matching a polyline perceived in front of the robot to a polyline that is based on the estimated position of the robot to its right, the robot must have turned right. Hence, we demand all induced alignments to be alike. To enable efficient computation of the matching, an estimation of the induced alignment is required. It can either be derived from odometry (if available) or simply reflect the assumption that the robot has not moved. We stress that we do not use any odometry data in our approach.

Let us assume that $\vec{B} = (B_1, B_2, \ldots, B_b)$ and $\vec{B'} = (B'_1, B'_2, \ldots, B'_{b'})$ are two cyclic ordered vectors of polylines. Denoting correspondence of B_i and B'_j by relation $\sim$, the task can be formulated as minimization as follows.

$$\sum_{(\vec{B}_i, \vec{B'}_j) \in \sim} S(\vec{B}_i, \vec{B'}_j) + C \cdot (2|\sim| - |\vec{B}| - |\vec{B'}|) \stackrel{!}{=} \min$$

Hereby, C denotes a penalty for not matching a polyline. This is necessary, as not establishing any correspondence would otherwise yield the lowest possible similarity 0. The similarity measure S is composed of the shape similarity measure presented in Section 3.2 and the alignment measure considering the difference between the alignment induced by the corresponding polylines' and the estimated one.

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

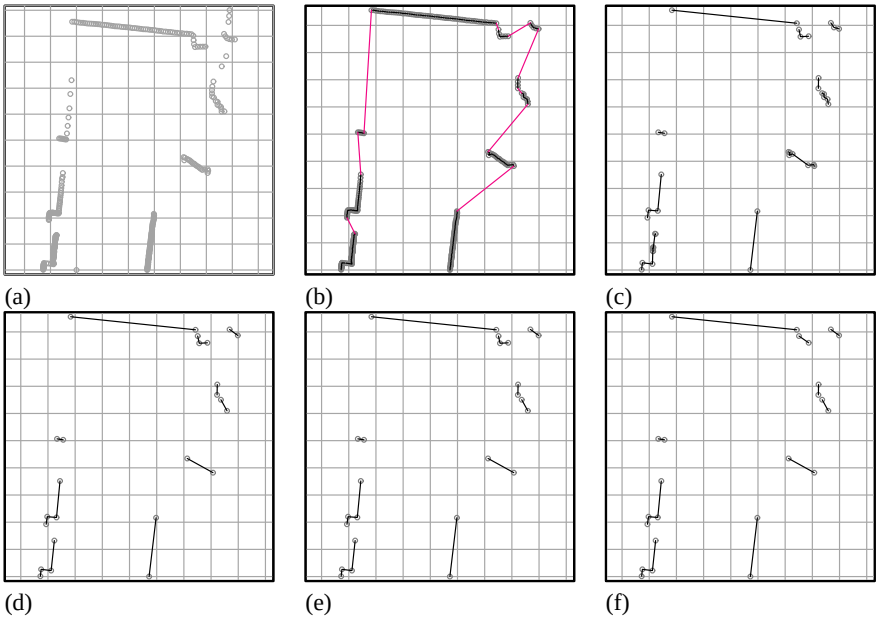

Figure 2. The process of extracting polygonal features from a scan. Raw scan points (a) are grouped to polylines (b), then simplification by means of DCE is performed. Figures (c) to (f) show various simplification levels (1,5,10, and 15 respectively) highlighting that the precise choice the simplification level is not critical for shape information obtained. The grid denotes 1 meter distance.

Considering alignments becomes necessary when many featureless shapes, e.g., chairs' legs, need to be tracked.

We use an adequate extension of the dynamic programming scheme to compute the best correspondence. The extension regards the ability to detect even 1-to-many and many-to-1 correspondences of polylines and results in a linear extra effort such that the overall complexity is $O(n^3)$. Observe that n is very low, it was around 10 for our example map, since this is the number of polylines in a scan. The outlined matching is powerful enough to track shapes even if no odometry information is available and the robot has traveled a remarkable distance between two consecutive scans. Figure 3 shows the polyline correspondence computed by our method, where the corresponding polylines from two different scans are connected by dashed lines. As can be observed, approaches based on the nearest point rule fail in this case.

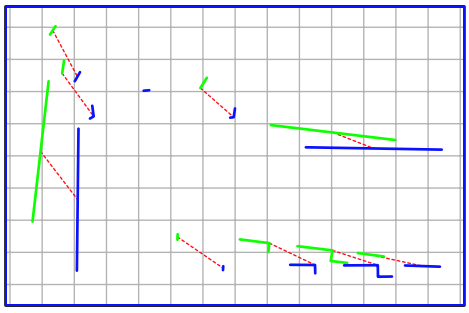

Figure 3. Exemplary results of the shape based matching for two scans (green and blue polylines; the grid denotes 1m distance).

4 MERGING

Merging an aligned new scan to the global map adds newly detected features in the surrounding area to the global map while not discarding the existing features no longer seen by the robot. It produces a global map composed of polylines.

It is common that portions of the same object may be perceived as several independent objects in a single scan. Such a phenomenon can happen when there is another object blocking the view, or simply because of the angle and the distance from which the object is viewed. During the merging process, we need to accurately identify objects in each scan, look for their corresponding objects in the existing global map, calculate the updated position of the object, remove moving objects, remove areas where there are objects with nonrigid shape, and more importantly, merge several objects into a single object whenever possible.

The main idea of the presented merging process is to simulate the robot laser scanner to merge the aligned scan S_t to the global map G_{t-1}. As output we obtain an updated global map G_t. The simulated laser rays emanate from the current robot pose and follow the parameters of the robot scan device. This means for our data that the simulated rays move counter-clockwise in steps of 0.5° and cover the angle of 180°. If a ray intersects a polyline in S_t, we select the closest intersection point to the robot pose, which is called a simulated scan point (ssp). We are creating at most 361 ssps. Furthermore, we add all end points of polylines in S_t to the list of newly created scan points with proper ordering (since we know between which scan rays they lie). From each ssp p_s in S_t, we find the closest point p_g on G_{t-1}. If the distance from p_s to p_g is below a certain threshold, then we take a weighted average of the two points to create a new point for the new global map G_t. We use a larger weight for the point p_g in G_{t-1}, since we have more confidence about its position. If the closet point cannot be found within a certain distance, the ssp p_s will be a new point on the new global map.

Since we want every polyline vertex in a global map G_{t-1} to have a corresponding point in the new global map, we have the following rule. If two consecutive ssps p_{s_1} and p_{s_2} have two corresponding closest points p_{g_1} and p_{g_2} in G_{t-1}, and there is a vertex v between p_{g_1} and p_{g_2}, we create a new ssp p_{s_v} in S_t that is the closest point

to v in the part of S_t between p_{s_1} and p_{s_2}. Then we take a weighted average of the two points p_{s_v} and v to create a new point for the new global map G_t. This step is illustrated in Figure 4.

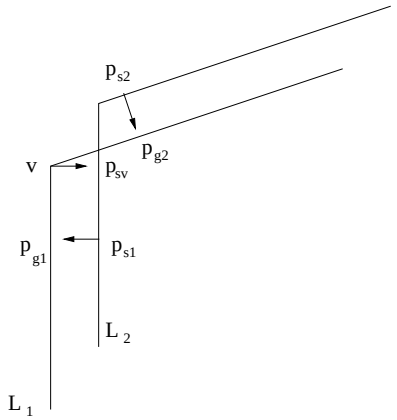

Figure 4. v is a vertex of a polyline L_1 of the global map G_{t-1}. v is between the closest points p_{g_1} and p_{g_2} to two consecutive simulated scan points (ssps) p_{s_1} and p_{s_2} on polyline L_2 in S_t. In such case, we create a new ssp p_{s_v} in the scan S_t as the closest point to v on a polyline L_2 of the scan S_t. Then we take a weighted average of the two points p_{s_v} and v to create a new point for the new global map G_t.

The next step is to create a set of ordered polylines from the newly created points (ncps) in the global map G_t. The following facts guide this process:

1. All ncps in the global map G_t are ordered. They either inherited their order from the simulated scan rays or where sorted in between two ssps (who inherited their order from the simulated scan rays).
2. All consecutive ncps whose predecessors belonged to the same polyline either in G_{t-1} or in S_t are classified as belonging to the same polyline.
3. The parts of polylines in G_{t-1} that are not predecessors of any ncps are integrated in G_t as separate polylines.

First we connect all consecutive (in scan order) ncps in the new global map that belong to the same polyline. We further merge consecutive polylines P_1 and P_2 to a single polyline Q if the last vertex of P_1 and the first vertex of P_2 had predecessors in the same polyline either in S_t or in G_{t-1}. The motivation for this step is that the polyline Q is likely to represent a single object in G_t (created from S_t and G_{t-1}), since whenever two consecutive points are in the same object either in S_t or in G_{t-1}, they can be classified as belonging to the same object. This rule implies that separate portions of the same object will be connected if they are ever detected as points of the same object. The only exception to this rule is a bifurcation, which may be caused by dynamic object, e.g., a moving door. To resolve bifurcations, we create two disjoint polylines in a global map.

The created polylines in the new global map G_t may not be properly ordered. A polyline P is *properly ordered* if the order of its vertices is constant with the arc length distance to its first vertex a i.e., vertex v proceeds w if the arc length distances satisfy $d_P(a,v) < d_P(a,w)$. We apply recursively the following simple rule to obtain properly ordered polylines: Let u, v, w be three consecutive vertices in a polyline P, if the inner angle at v is less then a certain threshold (which is 20^o in our case), then vertex v is removed. This rule not only makes polylines properly ordered, but also removes scan artifacts due to sensor noise or due to objects with fine shape features like plants. This rule is justified by the fact that sharp inner angles cannot be created by three consecutive laser rays unless they hit an object with fine shape.

For objects with fine shape, we obtain a dense sequence of sharp inner angles. Since this rule would remove them completely, these kind of objects requires a special treatment. Such objects have different reflections patters for different scans, thus, their shape may change from scan to scan. These object are not considered in the merging process, but they are placed on the global map after merging. To (temporarily) remove objects, we use a bounding box around them, called "eraser box", to erase the areas where the perceived data is highly unreliable. We create an eraser box in a scan by creating a bounding rectangle containing all consecutive vertices in each polyline of S_t with inner angles that are less than $20°$. If such a sharp angle is span by three consecutive scan points, it is very unlikely that these points result from scan readings of a real object with a stable shape. Thus, such sharp angles indicate either noisy readings or an object with fuzzy shape, like plants. The processing of eraser boxes is composed of three main steps: (1) for each scan we identify eraser boxes, (2) we transfer the eraser boxes to the global map, and (3) we merge overlapping eraser boxes in the global map to a single eraser box that contain them. We need these three steps, because the data in a small area may appear to be reliable in a given scan, but appears highly unreliable in most of the other scans. Thus, we mark the whole area unreliable regardless of the quality of any single scan. The data inside the eraser boxes on the global map is not used for further iterations of building the global map.

In order to create stable global maps, we can only merge perceived stationary objects, i.e., we need to detect moving objects and remove them from actual scans before merging. To remove moving objects, we use the result of shape matching to identify corresponding objects in scan S_{t-1} and S_t, say O_{t-1} is matched to O_t. We can also easily remember in the process of merging that O_{t-1} is merged into the object M_{t-1} in G_{t-1}. Since O_t and M_{t-1} represent the same real object, their distance after alignment should be very small. If this is not the case, the object O_t is moving, in which case we need to backtrack to its first appearance in the scans, remove it, and redo the merging process again to remove any unwanted effects it may have caused.

Finally, we apply DCE and least square fitting to create a new set of simplified polylines. An example result is shown in Fig. 1(d).

5 CONCLUSIONS

We have presented a comprehensive geometric model for robot mapping based on shape information. Shape matching has been tailored to the domain of scan matching. The matching is powerful enough to disregard pose information and cope with significantly differing scans. This improves performance of today's scan matching approaches dramatically. In this paper we concentrate on solving geometric problems related to global map building, and do not include any statistical methods, to demonstrate the power of the proposed geometric approach. However, we are aware that statistical methods are needed to guarantee robust performance. A suitable extension of our approach to include statistical methods will be presented in a separate paper.

ACKNOWLEDGEMENTS

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. It was carried out in collaboration with the SFB/TR 8 Spatial Cognition, project R3 [Q-Shape] support by the Deutsche Forschungsgemeinschaft (DFG). All support is gratefully acknowledged.

REFERENCES

[1] M. Arkin, L. P. Chew, D. P. Huttenlocher, K. Kedem, and J. S. B. Mitchell (1991). An efficiently computable metric for comparing polygonal shapes. *IEEE Trans. PAMI*, 13:209–206.

[2] Cox, I.J., Blanche – An experiment in Guidance and Navigation of an Autonomous Robot Vehicle. *IEEE Transaction on Robotics and Automation* 7:2, 193–204, 1991.

[3] Dissanayake, G. ,Durrant-Whyte, H., and Bailey, T., A computationally efficient solution to the simultaneous localization and map building (SLAM) problem. *ICRA'2000 Workshop on Mobile Robot Navigation and Mapping*, 2000.

[4] J. Forsberg, U. Larsson, and A. Wernersson. Mobile Robot Navigation using the Range-Weighted Hough Transform. *IEEE Robotics and Automation Magazine* 21, pp. 18-26, 1995.

[5] Gutmann, J.-S., Schlegel, C., AMOS: Comparison of Scan Matching Approaches for Self-Localization in Indoor Environments. *1st Euromicro Workshop on Advanced Mobile Robots (Eurobot)*, 1996.

[6] Gutmann, J.-S. and Konolige, K., Incremental Mapping of Large Cyclic Environments. *Int. Symposium on Computational Intelligence in Robotics and Automation (CIRA'99)*, Monterey, 1999.

[7] Gutmann, J.-S., Robuste Navigation mobiler System, *PhD thesis*, University of Freiburg, Germany, 2000.

[8] D. Hähnel, D. Schulz, and W. Burgard. Map Building with Mobile Robots in Populated Environments, *Int. Conf. on Int. Robots and Systems (IROS)*, 2002.

[9] P. Jensfelt and H. I. Christensen. Pose Tracking Using Laser Scanning and Minimalistic Environmental Models, *IEEE Trans. on Robotics and Automation*, 2001.

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

[11] L. J. Latecki and R. Lakämper (1999). Convexity Rule for Shape Decomposition Based on Discrete Contour Evolution. *Computer Vision and Image Understanding* 73:441–454.

[12] L. J. Latecki and R. Lakämper (2000). Shape Similarity Measure Based on Correspondence of Visual Parts. *IEEE Trans. Pattern Analysis and Machine Intelligence* 22:1185-1190.

[13] Lu, F., Milios, E. (1997). Robot Pose Estimation in Unknown Environments by Matching 2D Range Scans. *Journal of Intelligent and Robotic Systems* 18:3 249–275

[14] Röfer, T., Using Histogram Correlation to Create Consistent Laser Scan Maps . *IEEE Int. Conf. on Robotics Systems (IROS)*. Lausanne, Switzerland, 625–630, 2002.

[15] Thrun, S. (2002). Robot Mapping: A Survey, In Lakemeyer, G. and Nebel, B. (eds.): *Exploring Artificial Intelligence in the New Millenium*, Morgan Kaufmann, 2002.