Shape Similarity and Visual Parts

Longin Jan Latecki[1], Rolf Lakämper[1], and Diedrich Wolter[2]

[1] Dept. of Computer and Information Sciences, Temple University
Philadelphia, USA
{latecki,lakamper}@temple.edu

[2] Dept. of Computer Science, University of Bremen, Bremen, Germany
dwolter@informatik.uni-bremen.de

Abstract. Human perception of shape is based on visual parts of objects to a point that a single, significant visual part is sufficient to recognize the whole object. For example, if you see a hand in the door, you expect a human behind the door. Therefore, a cognitively motivated shape similarity measure for recognition applications should be based on visual parts. This cognitive assumption leads to two related problems of scale selection and subpart selection. To find a given query part Q as part of an object C, Q needs to have a correct size with regards to C (scale selection). Assuming that the correct size is selected, the part Q must be compared to all possible subparts of C (subpart selection). For global, contour-based similarity measures, scaling the whole contour curves of both objects to the same length usually solves the problem of scale selection. Although this is not an optimal solution, it works if the whole contour curves are 'sufficiently' similar. Subpart selection problem does not occur in the implementation of global similarity measures.

In this paper we present a shape similarity system that is based on correspondence of visual parts, and apply it to robot localization and mapping. This is a particularly interesting application, since the scale selection problem does not occur here and visual parts can be obtained in a very simple way. Therefore, only the problem of subpart selection needs to be solved. Our solution to this problem is based on a contour based shape similarity measure supplemented by a structural arrangement information of visual parts.

1 Motivation and Overview of Shape Descriptors

Shape descriptors for comparing silhouettes of 2D objects in order to determine their similarity are important and useful for wide range of applications, of which the most obvious is shape-based object retrieval in image databases. Shape's importance is indicated by the fact that the MPEG-7 group incorporated shape descriptors into the MPEG-7 standard. Since the 2D objects are projections of 3D objects their silhouettes may change due to:

1. change of a view point with respect to objects,
2. non-rigid object motion (e.g., people walking or fish swimming),
3. noise (e.g., digitization and segmentation noise).

I. Nyström et al. (Eds.): DGCI 2003, LNCS 2886, pp. 34–51, 2003.

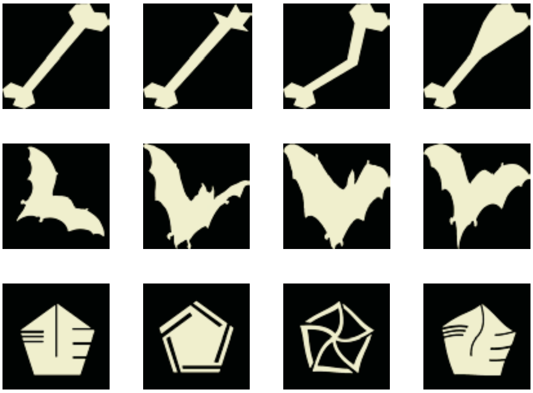

Fig. 1. Some shapes used in part B of MPEG-7 Core Experiment CE-Shape-1. Shapes in each row belong to the same class.

The goal of the Core Experiment CE-Shape-1 [20] was to evaluate the performance of 2D shape descriptors under such conditions. The shapes were restricted to simple pre-segmented shapes defined by their bitmaps. Some example shapes are shown in Figure 1. The main requirement was that the shape descriptors should be robust to small non-rigid deformations due to (1), (2), or (3). In addition the descriptors should be scale and rotation invariant.

The main part of the Core Experiment CE-Shape-1 was part B: similarity-based retrieval. The data set used for this part is composed of 1400 shapes stored as binary images. The shapes are divided into 70 classes with 20 images in each class. In the test, each image was used as a query, and the number of similar images (which belong to the same class) was counted in the top 40 matches (bulls-eye test). Since the maximum number of correct matches for a single query image is 20, the total number of correct matches is 28000.

It turned out that this data set is the only set that is used to objectively evaluate the performance of various shape descriptors. We present now some of the shape descriptors with the best performance on this data set. It is not our goal to provide a general overview of all possible shape descriptors. A good overview can be found in the book by Costa and Cesar [4].

The shape descriptors can be divided into three main categories:

1. *contour based descriptors:* the contour of a given object is mapped to some representation from which a shape descriptor is derived,
2. *area based descriptors:* the computation of a shape descriptor is based on summing up pixel values in a digital image of the area containing the silhouette of a given object; the shape descriptor is a vector of a certain number of parameters derived this way (e.g., Zernike moments [13]),
3. *skeleton based descriptors:* after a skeleton is computed, it is mapped to a tree structure that forms the shape descriptor; the shape similarity is computed by some tree-matching algorithm.

The idea of representing shapes by their skeletons in Computer Vision goes back to Blum [3]. Siddiqi et al. [25] also convert object skeletons to a tree representation and use a tree-matching algorithm to determine the shape similarity.

In the MPEG-7 Core Experiment CE-Shape-1 part B, shape descriptors of all three categories were used. A general conclusion is that contour based descriptors significantly outperformed the descriptors of the other two categories [20]. It seems to be that area based descriptors are more suitable for shape classification than for indexing. The week performance of skeleton based descriptors can probably be explained by unstable computation of skeletons related to discontinuous relation between object boundary and skeletons. A small change in the object boundary may lead to a large change in the skeleton.

As reported in [20], the best retrieval performance of 76.45% for part B was obtained for shape descriptor of Latecki and Lakaemper [17], that will be described in this paper, (presented by the authors in cooperation with Siemens Munich) followed by shape descriptor of Mokhtarian et al. [22,23] with retrieval rate of 75.44% (presented by Mitsubishi Electric ITE-VIL). It is important to mention that 100% retrieval rate on this data set is not possible to achieve employing only shape. The classification of the objects was done by human subjects, and consequently, some shapes can be only correctly classified when semantic knowledge is used.

Meanwhile new shape descriptors have been developed that yield a slightly better performance. The best reported performance on this data set is obtained by Belongie et al. [2], 76.51%. The small differences in the retrieval rate of these approaches are more likely to indicate a better parameter tuning than a better approach.

All the contour based shape descriptors have a common feature that limits their applicability. They require a presence of the whole contours to compute shape similarity. Although they are robust to some small distortions of contours, they will fail if a significant part of contour is missing or is different. The same critique applies to area and skeleton based shape descriptors that require the whole object area or the complete skeleton to be present.

The goal of this paper is to direct our attention to a cognitively motivated ability of shape descriptors and the shape similarity measures that is necessary for most practical applications of shape similarity. It is the ability of partial matching.

Partial matching leads to two related problems of scale selection and subpart selection. To find a given query part Q as part of an object C, Q needs to have a correct size with regards to C (scale selection). Assuming that the correct size is selected, the part Q must be compared to all possible subparts of C (subpart selection). The subparts may be obtained either by a decomposition of Q into parts using some decomposition criterion or simply by sliding Q over all possible positions with respect to C, e.g., the beginning point of Q is aligned with each point of C.

A good example of an approach that allows for partial matching is a single-directional Hausdorff distance [12], which tries to minimize the distance of all points of the query part Q to points of object C. However, the problem of scale selection cannot be solved in the framework of Hausdorff distance alone. For example, the approach presented in [12] simply enumerates all possible scales. Moreover, the Hausdorff distance does not tolerate shape deformations that preserve the structure of visual parts, i.e., the objects differing by such deformations although very similar to humans will have a large similarity value.

For global, contour-based similarity measures, scaling the whole contour curves of both objects to the same length usually solves the problem of scale selection. Although this is not an optimal solution, it works if the whole contour curves are 'sufficiently' similar. Subpart selection problem does not occur in the implementation of global similarity measures.

To our knowledge, there does not exist an approach to partial shape similarity that also solves the scaling problem. In this paper we show that the shape descriptor presented by Latecki and Lakaemper [17] can be easily modified to perform partial matching when the scale is known. An ideal application where this restriction is satisfied is robot localization and mapping using laser range data. Therefore, we apply our shape similarity measure in this context.

2 Shape Representation, Simplification, and Matching

For a successful shape-representation we need to account for arbitrary shapes. Any kind of boundary information obtained must be representable. Therefore, we will use polygonal curves as boundary representation. We developed a theory and a system for a cognitively motivated shape similarity measure for silhouettes of 2D objects [17,18,16].

To reduce influence of digitization noise as well as segmentation errors the shapes are first simplified by a novel process of discrete curve evolution which we introduced in [16,19]. This allows us

- (a) to reduce influence of noise and
- (b) to simplify the shape by removing *irrelevant* shape features without changing *relevant* shape features.

A few stages of our discrete curve evolution are shown in Figure 2. The discrete curve evolution is context sensitive, since whether shape components are relevant or irrelevant cannot be decided without context. In [16], we show that the discrete curve evolution allows us to identify significant visual parts, since significant visual parts become maximal convex arcs on an object contour simplified by the discrete curve evolution.

Let P be a polyline (that does not need to be simple). We will denote the vertices of P by $Vertices(P)$. A *discrete curve evolution* produces a sequence of polylines $P = P^0, ..., P^m$ such that $|Vertices(P^m)| \leq 3$, where $|\ .\ |$ is the cardinality function. Each vertex v in P^i (except the first and the last if the polyline is not closed) is assigned a relevance measure that depends on v and its two neighbor vertices u, w in P^i:

$$K(v, P^i) = K(u, v, w) = |d(u, v) + d(v, w) - d(u, w)|, \qquad (1)$$

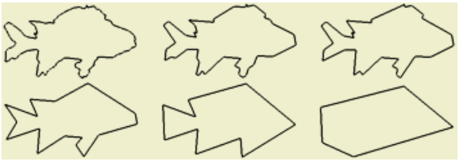

Fig. 2. A few stages of our discrete curve evolution.

where d is the Euclidean distance function. Note that K measures the bending of P^i at vertex v; it is zero when u, v, w are collinear.

The process of *discrete curve evolution (DCE)* is very simple:

- At every evolution step $i = 0, ..., m-1$, a polygon P^{i+1} is obtained after the vertices whose relevance measure is minimal have been deleted from P^i.

For end vertices of open polylines no relevance measure is defined, since the end vertices do not have two neighbors. Consequently, end-points of open polylines remain fixed.

Note that P^{i+1} is obtained from P^i by deleting such a vertex that the length change between P^i and P^{i+1} is minimal. Observe that relevance measure $K(v, P^i)$ is not a local property with respect to the polygon $P = P^0$, although its computation is local in P^i for every vertex v. This implies that the relevance of a given vertex v is context dependent, where the context is given by the adaptive neighborhood of v, since the neighborhood of v in P^i can be different than its neighborhood in P. The discrete curve evolution has also been successfully applied in the context of video analysis to simplify video trajectories in feature space [6,15].

DCE may be implemented efficiently. Polyline's vertices can be represented within a double-linked polyline structure and a self-balancing tree simultaneously. Setting up this structure for a polyline containing n vertices has the complexity of $O(n \log n)$. A step within DCE constitutes of picking out the least relevant point ($O(\log n)$), removing it ($O(\log n)$), and updating it's neighbor's relevance measures ($O(1)$). As there are at most n points to be deleted, this yields an overall complexity of $O(n \log n)$. As it is applied to segmented polylines, the number of vertices is much smaller than the number of points read from the sensor.

To compute our similarity measure between two polygonal curves, we establish the best possible correspondence of maximal convex arcs. To achieve this, we first decompose the polygonal curves into maximal convex subarcs. Since a simple one-to-one comparison of maximal convex arcs of two polygonal curves is of little use, due to the facts 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 the maximal convex 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 conour composed of adjacent

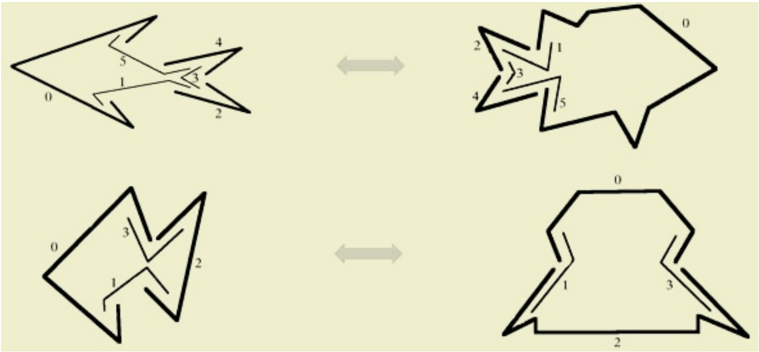

Fig. 3. The corresponding arcs are labeled by the same numbers.

maximal convex arcs. In this context the corresponding parts of contours can be identified with visual object parts. The best correspondence of the visual object parts, 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 programing, the similarity between corresponding parts is computed and aggregated. The computation is described extensively in [17]. The similarity induced from the optimal correspondence of polylines C and D will be denoted $S(C, D)$. Two example correspondences obtained by our approach are shown in Fig. 3. Since our shape matching technique is based on correspondence of visual parts, it will also work under a moderate amount of occlusion and/or segmentation errors.

Basic similarity of arcs is defined in tangent space. Tangent space, also called *turning function*, is a multi-valued step function mapping a curve into the interval $[0, 2\pi)$ by representing angular directions of line-segments only. Furthermore, arc lengths are normalized to 1 prior to mapping into tangent space. This representation was previously used in computer vision, in particular, in [1]. Denoting the mapping function by T, the similarity gets defined as follows:

$$S_{arcs}(C, D) = \left(\int_0^1 (T_C(s) - T_D(s) + \Theta_{C,D})^2 ds \right) \cdot \max \left\{ \frac{l(C)}{l(D)}, \frac{l(D)}{l(C)} \right\}, \qquad (2)$$

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

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

Obviously, the similarity measure is a rather a dissimilarity measure as the identical curves yield 0, the lowest possible measure. 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 closed contours.

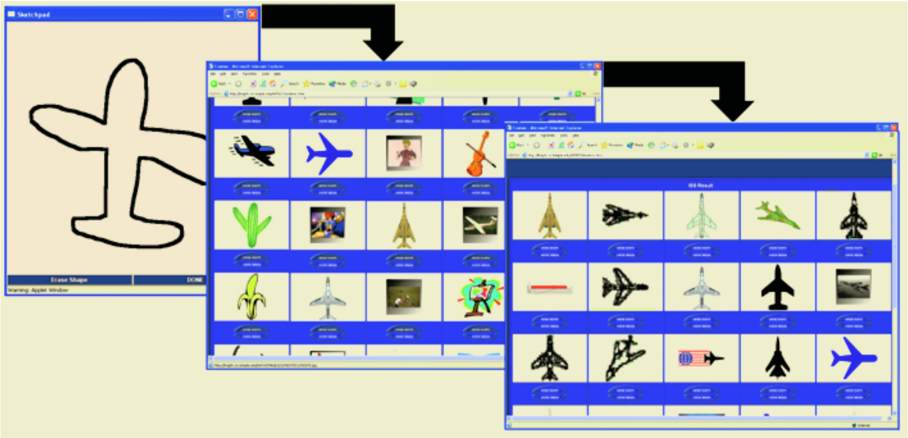

Fig. 4. An illustration of query process in our shape database; from left to right: query sketch, first result, and refined result.

3 First Application: Image Database

The performance of our shape descriptor (described in Section 2) can be evaluated using the shape-based image database located at

`http://knight.cis.temple.edu/~shape`

The interface allows query by shape based on hand-drawn sketches as well as texture and keywords. Using shape, the user defines the query drawing a shape boundary, see Figure 4(left). Since the system has to deal with moderate artistic abilities of the user (who may not be a gifted illustrator) the results are achieved in two steps of increasing precision: the first result set shows examples of different shape classes, presenting not a precise match but a wide variety of similar shapes. The reason is that not all parts existing in the hand-drawn sketch are considered as descriptive features. A typical example is an airplane roughly drawn from top view: the first search result includes planes, but also shows a cactus, a peeled banana etc., Figure 4(middle); note that these shapes have a similar boundary to a plane.

To refine the search, one of the shapes is chosen as new query, which is now an object formerly stored in the database. It is independent from the user's sketching talents, therefore it is reasonable to enhance the search precision based on all parts of the shape. The results of this second query are the most similar matches in the database using our similarity measure. The shapes in Figure 4(right) are the best matches for the airplane in the center of first result.

The search can be recursively continued by choosing shapes of each result set as new query. Since the boundary of the chosen shape is first imported into the input-interface, it is possible to further enhance the search by additional information (e.g. texture).

4 Second Application: Robot Mapping and Localization

Robot mapping and localization are the key points in building truly autonomous robots. The central method required is 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 [28], Dissanayake et al. [7]). The extended Kalman filter, a linear recursive estimator for systems described by non-linear process models, and observation models are usually employed in current SLAM algorithms.

The robot's internal geometric representation builds the basis for these techniques. It is build atop of the perceptual data read from the laser range finder. Typically, either the planar location of reflection points read from the laser range finder is used directly as the geometric representation, or simple features in form of line segments or corner points are extracted (Cox [5]; Gutmann and Schlegel [8]; Gutmann [10]; Röfer [24]). Although robot mapping and localization techniques are very sophisticated they do not yield the desired performance in all respects. We observe that these systems use only a very primitive geometric representation. As the internal geometric representation is the foundation for localization and mapping, shortcomings on the level of geometric representation affect the overall performance.

Systems with geometric representation based on the extracted features outperform the systems based on the location of scan points in terms of compute time, but there is a major drawback. Systems relying on linear features can only cope with surroundings that are largely made up from linear segments. Hence, these approaches are limited to indoor office scenarios (Röfer [24]). To cope with unconstrained scenarios as needed for service robot applications, more general features are required, as most environments, like furnished rooms, lack of linear features but show a great variety of shapes. Figure 5 gives an impression of a regular home scenario. Furthermore, extracting lines from an environment lacking of exactly linear parts but presenting many slightly curved ones introduces a lot of noise. This noise affects the matching quality. As this noise is propagated from matching to matching, it accumulates, resulting in errors. But just like environments lacking of the features chosen for mapping, the presence of a lot of those features can lead to difficulties. Problems arise in a surrounding containing many similar features. For example, scanning a zigzag- shaped wall (or a curtain) results in detecting many lines at positions nearby each other pointing in similar directions. Applying a line-based matching treats all lines individually, a matching is susceptible to a mix-up. Hence, the map gets misaligned.

Besides the specific shortcomings discussed, it has been claimed by various authors that using purely metric geometric representation will not suffice for a mobile robot system. Especially solving navigational tasks can benefit from a more abstract representation, e.g. a topological one. As metric information is needed for path planning and topological information is desired in navigation,

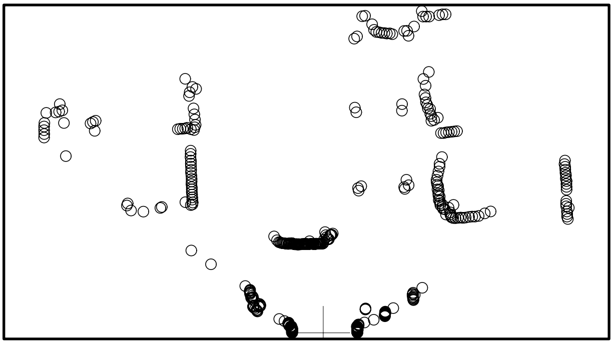

Fig. 5. A regular living room perceived by a laser range finder. Each circle represents a reflection point measured. The lack of linear features is evident. Hence, more complex, versatile features need to be employed. The cross denotes the position and the orientation of the robot.

an urge to abstract from metrical data arises. Therefore, hybrid representations have been proposed (Thrun [26]; Kuipers [14]). Thus, a representation granting topological access alongside the metric information would be advantageous.

Using either a feature extraction or not, mapping applications are opposed with another problem yet. As a topological correct map is a prerequisite to a successful navigation, maintaining topological correctness is a key point. We discuss two problems that require a careful mapping in order not to violate topology. The first problem is with self-intersections. Among existing approaches there are no global geometric constraints that prevent the resulting map from containing any overlaps. Such overlaps between parts of the map wrongly restrict the robot's passable space. Maps containing such errors can no longer be used for navigation. The second problem is the cycle detection. The problem is illustrated in Figure 6(a).

To link processing of perceptual data and handling of navigational tasks more fitting together, we believe introducing an improved geometric representation as basis of a mobile robot's spatial representation is the central point. A successful geometric representation must result in a much more compact representation than uninterpreted perceptual data, but must neither discard valuable information nor imply any loss of generality. We claim that a shape-representation as the robot's underlying spatial representation fulfills these demands. Representing the passable space explicitly by means of shape is not only adequate to mapping applications but helps also to bridge the gap from metric to topological information due to the object-centered perspective offered. Moreover, an object-centered representation is a crucial building block in dealing with changing environments, as this representation allows us to separate the partial changes from the unchanged parts.

The demands posed on a scan matching are similar to the ones in computer vision as discussed in the beginning: the environment is perceived from different view points, the environment is composed of different visual parts, and sensor

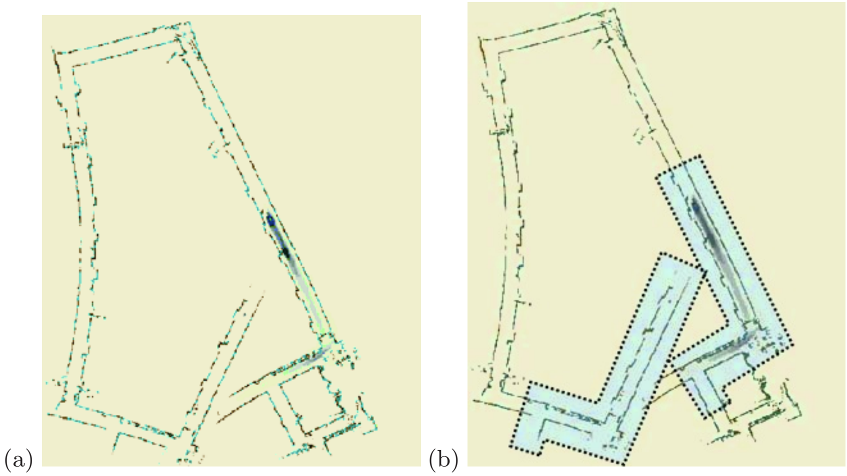

Fig. 6. (a) This figure from the paper by Gutmann and Konolidge [9] shows a partially mapped environment. Due to the propagation of errors, the cyclic path the robot was following is no longer cyclic. Subsequent mapping would lead to an overlap. (b) Using shape-similarity we can detect the overlapping parts (highlighted).

data is noisy. This provides a strong connection to shape matching. Although it has been stated in Lu and Milos' fundamental work [21], *"scan matching is similar to model-based shape matching"*, approaches to scan-matching have so far not taken advantage of state-of-the-art techniques in shape-matching.

We propose a shape-representation of the robot's surrounding as it is perceived by a laser range finder (LRF). After the scan points are mapped to a 2D top view of the surrounding, they can be easily grouped to form connected polylines. Our features are these polylines, which we interpret as visual parts of the boundary of the scanned objects. Shape processing and matching of these visual parts allow us to derive a sophisticated matching of scans that is reliable as well as efficient. Using visual parts as features allows us to maintain the generality required for arbitrary indoor scenarios, since the boundary of any shape can be easily represented with a polyline. 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.

Our motivation for this approach is related to the human visual perception, where shape representation and recognition plays a primary role. It is well-known that it is the case in object recognition. We claim that it is also the case for localization tasks and for route description in navigation.

In the following part of this paper, we will show that the proposed shape-based representation and matching of LRF scans lead to robust robot localization and mapping. Moreover, shape matching allows us to also perform object recognition (as it is the case in Computer Vision). This ability is extremely useful to maintain the global map consistency in robot navigation as we illustrate on the problem of cycle detection now. Using shape representation and shape similarity measure we can easily correct the map depicted in Figure 6(a). A shape matching procedure can identify that the two parts marked in Figure 6(b) are very similar. Since these parts have a complicated shape structure, the probability of an accidental similarity is very close to zero. By transforming the map so that the matching parts are aligned, we correct the map. Observe that this process is cognitively motivated, since a human observant will notice that the map in Figure 6(a) is incorrect and will correct it by identifying the highlighted parts in Figure 6(b) as having identical shape.

5 From LRF Scan Data to Simplified Polylines

This section details how boundary information of scanned objects is extracted from LRF data and how a similarity between two boundaries is determined. First the range data acquired by the laser range finder is mapped to locations of reflection points in the Euclidean plane, i.e., reflection points are represented as points in the plane. Thus, we obtain a sequence of scan points in the plane in a local coordinate system, the robot's heading aligned with the positive y-axis, e.g., see Figure 5. The order of the sequence reflects the order of the data as returned by the LRF.

The next step is to segment this sequence into polylines that represent visual parts of the scan. It must be noticed that this is necessary, since two consecutive points in the scan reading do not necessarily belong to the same object. In this case they must not be represented by the same polyline. For this segmentation, a simple heuristic may be used: Whenever the Euclidean distance of two consecutive points exceeds a given threshold (20 cm is used), these points are supposed to belong to different objects. The obtained polylines that represent boundaries of these objects are viewed as visual parts of the scan boundary. Thus, the extraction of visual parts in this context is a very simple process.

Segmented polylines still contain all the information read form the LRF. However, this data contains some noise. Therefore, we apply DCE (Section 2) that cancels noise as well as makes the data compact without loosing valuable shape information. To illustrate the complete process of feature extraction and, most importantly, the applicability of DCE to range finder data, refer to Figure 7.

Once the simplified boundaries are computed, a similarity of boundaries can be computed as described in Section 2. However, for matching two scans we will not rely only on matching individual boundaries. A structural shape representation representing all boundaries within a single compound object is used to avoid faulty matches.

6 Structural Shape Representation and Matching

The boundary-based computation of similarity provides a distinctive measure for matching boundaries against each other. However, self-similarities in the

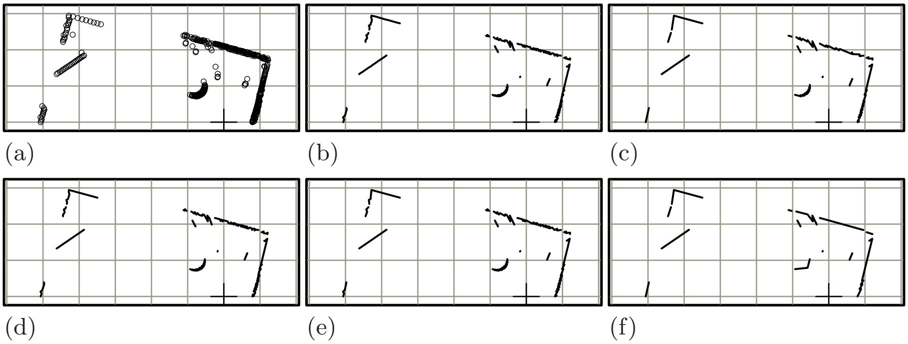

Fig. 7. The process of extracting polygonal features from a scan consists of two steps: First, polygonal lines are set up from raw scanner data (a) (1 meter grid, the cross denotes the coordinate system's origin). The lines are split, wherever two adjacent vertices are too far apart (20 cm). The resulting set of polygonal lines (b) is then simplified by means of discrete curve evolution with a threshold of 50. The resulting set of polygonal lines (c) consists of less data though still capturing the most significant information. Below, results of applying DCE with different parameters as threshold are shown. As can be observed, choosing the value is not critical for shape information. Thresholds chosen: (d) 10, (e) 30, (f) 70.

environment can still cause faulty matches. For example, within typical indoor scenarios, especially office buildings, there is a high self-similarity of objects, e.g., door frames look always the same. Though door frames can – due to their complexity – provide a distinctive shape feature, they might easily be mixed up when several of them are observable from a single viewpoint. Matching structural shape representations made up from an entire laser scan allows us to overcome this problem.

Structural representations allow us to incorporate abstract, qualitative knowledge (here: ordering information) and metric information into a single representation. Boundaries extracted from sensory data provide metric information needed for aligning scans. Bringing more abstract spatial knowledge into play enables an efficient matching. Just as the representation constitutes of two individual aspects, the employed shape-matching is likewise twofold. Matching shapes is build up from a matching of the shape's structure and from determining the similarity of boundaries. A similarity measure determined for a pair of boundaries – which were extracted from different scans – serves as a plausibility measure for the matching. The more similar these boundaries are, the more likely they correspond to each other.

The key point of the proposed approach is to dispose as much context information as possible. We elaborate on this in a bit more detail. Looking at a purely data-driven approach, there is no context information used at all. Each reflection point measured by the laser range finder is matched individually against another point from a different scan. Of course, such an attempt is prune to errors. Therefore, several enhancements need to be applied. The technique of position filtering is applied to neglect any reflection points in the matching process that

– most likely – could not have been observed from both viewpoints. Second, the displacement that aligns two scans best is determined as the mean value of the largest cluster of displacement induced from the individual correspondences of scan points[1].

Employing a feature-based scan-matching can be viewed as increase in context information. The local context of scan-points which form features is respected. However, features are still matched independently. Hence, computing the overall displacement still requires to compute a mean value from the most plausible cluster of individual displacements. The advantages of increasing the context respected in a matching can be summarized as (a) a reduction of compute time as there are just few features as compared to the raw LRF data and (b) an increase in matching reliability as a faulty correspondence of features is much more unlikely to happen accidentally as opposed to using raw data. Therefore, a structural shape representation is employed that captures the configuration of boundaries. Within the terminology of context, individual visual parts are matched in the context of a complete scan. This prevents mixups in determination of correspondence.

The aspect of spatial information stored in the structural representation is a very simple, yet powerful one: ordering information. Sensing a scene in a counter-clockwise manner induces a cyclic ordering of the objects perceived. When matching two scenes against each other, hence determining a correspondence of objects present in both scenes, this ordering must not be violated. As a LRF does not provide a full round view, the cyclic ordering may be represented by a linear ordered structure, i.e., a vector. Proceeding this way, we can represent a scan by a vector of visual parts (represented as boundary polylines) $\boldsymbol{B}$.

When matching two vectors of visual parts against each other, only 1-to-1-correspondences of boundaries are considered, but some visual parts may remain unmatched (new objects may appear and some objects may not longer be visible). Let us assume that all similarities for individual pairs of visual parts $S(B_i, B'_j)$ have been computed for two vectors $\boldsymbol{B} = (B_1, B_2, \ldots, B_b)$ and $\boldsymbol{B'} = (B'_1, B'_2, \ldots, B'_{b'})$ respectively, using our shape similarity measure S. Correspondence of visual parts B_i and B'_j will be denoted $B_i \sim B'_j$. Then the task to compute an optimal correspondence can be written as minimization of the summed up similarities $\mathit{\Sigma}_{(B_i,B'_j)\in\sim} S(B_i, B'_j)$. The goal is to compute the correspondence relation $\sim$ that yields the lowest overall sum of similarities of corresponding visual parts. To prevent a tendency not to match any visual parts (as $\sim = \emptyset$ would yield 0, the lowest sum possible), a penalty C is introduced for leaving a visual part unmatched, i.e., either $\forall i \in [1, \ldots, b']\boldsymbol{B}_i \not\sim \boldsymbol{B}'_j$ or $\forall j \in [1, \ldots, b]\boldsymbol{B}_i \not\sim \boldsymbol{B}'_j$. Thus, the matching can be written as minimization

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

[1] This can be viewed as introducing context information: the scan is treated as a compound object of points allowing scan-points only to be displaced equally.

Respecting the ordering of visual parts enforced by simply restricting the correspondence relation $\sim$ to be a strictly monoton ordering of indices i, j in $S(B_i, B'_j)$. Computing such optimal correspondence can be achieved by dynamic programming.

7 Aligning Scans

Once a correspondence has been computed, the scans involved need to be aligned in order to determine the current robot's position from which the latest scan has been perceived, and finally to build a global map from the perceived scans. To align two scans, a translation and rotation (termed a *displacement*) must be computed such that corresponding visual parts are placed at the same position. The overall displacement is determined from the individual correspondences. Of course, due to noise, this can only be fulfilled to a certain extend, as boundaries may sometimes not be aligned perfectly and individual displacements may differ. To define the best overall displacement, the overall error, i.e., the summed up differences to individual displacements, is minimized according to the method of least squares.

To mediate between all, possibly differing individual displacements, it is advantageous to restrict the attention to the most reliable matches. The presented approach uses only the best three matcheing pairs of visual parts selected using a reliability criterion described in Section 7.1.

Based on the correspondence of the three matcheing pairs two complete scan boundaries from time t and $t - 1$ are aligned. For each corresponding polyline pair, we also know the correspondence of the line segments of which the polylines are composed. These correspondences have been determined along the way of computing the similarity of two polylines. Proceeding this way, the problem of aligning two scan is reduced to aligning two sets of corresponding lines. This is tackled by computing the individual displacements that reposition the corresponding line segments atop each other using standard techniques. First, the induced rotation is computed as the average value of rotational differences and the scans are aligned accordingly. Second, the induced translation is computed. This is done by solving an over-determined set of linear equations. As due to noise usually no solution exists, the solution minimizing the least square error is chosen.

7.1 Matching Reliability

The reliability of a matching a pair of polylines is influenced by two parameters, namely their similarity and their *shape complexity.* The higher the complexity is, the more distinctive a matching is, as accidental matchings become much more unlikely with growing complexity. So, alongside the similarity measure complexity mirrors a plausibility for a particular matching. The motivation is that choosing the most complex correspondences from an overall matching of scans should guarantee to pick correct correspondences only. Determination of similarity measure S has been presented in section 2. To determine the complexity of a polyline P with points $(p_1, p_2, \ldots, p_n)$, $n > 2$ the following formula is used:

$$C_P = \sum_{i=2}^{n-1} K(p_{i-1}, p_i, p_{i+1}) \tag{3}$$

Hereby K denotes the relevance measure of points as defined in formula (1). For a polyline composed of a single line segment, however, no relevance measure can be assigned this way. Therefore, in this case simply the half length of the line segment is chosen as complexity (d denotes the Euclidean distance).

$$C_{(p_1,p_2)} = 0.5d(p_1, p_2) \tag{4}$$

The matching reliability of two polylines P, R is then determined by

$$Q(P, R) = C_P + C_R - S(P, R). \tag{5}$$

Thus, two polylines with complex shape that are very similar, receive a high matching reliability value.

7.2 Advanced Incremental Alignment

Previous sections explained how correspondences between two scans can be detected and how an induced displacement can be computed. In principle, an incremental scan matching can be realized in a straightforward manner: For each scan (at time t) visual parts are extracted and matched against the last scan perceived (at time $t - 1$). As the boundaries are matched they are displaced accordingly and entered in a map. However, such approach suffers from accumulating noise. For example, if a wall is perceived in front of the robot with a noise in distance of about $4cm$ (typical noise of a LRF), computing a single displacement can introduce an error of $8cm$. Such errors accumulate during the continuous matching. Hence, maps resulting from several hundred scans render themselves useless. This is reason enough for any real application to incorporate some handling of uncertainty, e.g., by means of stochastic models.

Our way of handling the uncertainty is again based on shape similarity. Instead of aligning all scans incrementally, i.e., scan t is aligned with respect to scan $t-1$, we align scan t with respect to a reference scan $t-n$ for some $n > 1$. Scan $t-n$ remains as the reference scan as long as the three most reliable maching visual parts from scan t are sufficiently similar to the corresponding visual parts from scan $t - n$. This reference scan allows us to keep the accumulating incremental error down, as the reference visual parts do not change so often. Our criterion on when to change the reference scan is a threshold on shape similarity of actual visual parts to the reference ones.

The performance of our system is demonstrated in Figure 8(a), where the map constructed from 400 scans obtained by a robot moving along the path marked with the dashed line is shown. For comparison, a ground truth map of the reconstructed indoor environment (a hallway at the University of Bremen) is shown in 8(b).

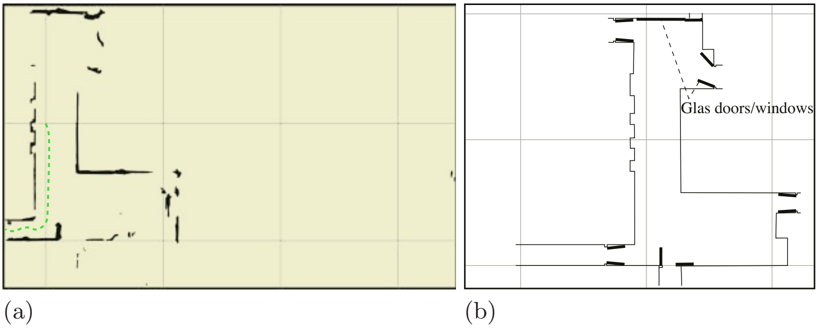

Fig. 8. (a) A map created by our approach. The robot path is marked with a dashed line. (b) A ground truth map of the indoor environment.

8 Conclusions

The problems of self-localization and robot mapping are of high importance to the field of mobile robotics. These problems constitute from a geometric level and a handling of uncertainty. State-of-the art in robot mapping and self-localization provides us with good techniques to master the latter. The underlying geometric representation is a rather simple one. Either perceptual data remains largely uninterpreted or simple features (e.g. lines, corners) are extracted. A connection between the geometric level and shape matching exists but is still underexploited. By using a shape representation as the underlying geometric representation, we combined advantages of feature-based approaches, namely a compact representation and a high-level, object-centered interface, with generality of uninterpreted approaches due to shape-representation's versatility.

Our future goal is to gain deeper geometric understanding of robot localization. It is well known that shape representation and shape-based object recognition plays a primary role in human visual perception. Our research indicates that localization and mapping tasks are also based on shape representation and shape matching. Therefore, we are developing a robot localization and mapping formalism that employs a cognitively motivated shape representation and shape matching.

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. S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 24:509–522, 2002.
3. H. Blum. Biological shape and visual science. *Journal of Theor. Biol.*, 38:205–287, 1973.

4. L. da F. Costa and R. M. Cesar. *Shape Analysis and Classification. Theory and Practice.* CRC Press, Boca Raton, 2001.
5. 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.
6. D.F. DeMenthon, L.J. Latecki, A. Rosenfeld, and M. Vuilleumier Stückelberg. Relevance ranking and smart fast-forward of video data by polygon simplification. pages 49–61, 2000.
7. 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.
8. 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.
9. 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.
10. Gutmann, J.-S., Robuste Navigation mobiler System, *PhD thesis*, University of Freiburg, Germany, 2000.
11. D. Hähnel, D. Schulz, and W. Burgard. Map Building with Mobile Robots in Populated Environments, *Int. Conf. on Int. Robots and Systems (IROS)*, 2002.
12. D. Huttenlocher, G. Klanderman, and W. Rucklidge. Comparing images using the Hausdorff distance. *IEEE Trans. PAMI*, 15:850–863, 1993.
13. A. Khotanzan and Y. H. Hong. Invariant image recognition by zernike moments. *IEEE Trans. PAMI*, 12:489–497, 1990.
14. B. Kuipers. The Spatial Semantic Hierarchy, *Artificial Intelligence* 119, pp. 191–233, 2000.
15. L. J. Latecki and D. de Wildt. Automatic recognition of unpredictable events in videos. In *Proc. of Int. Conf. on Pattern Recognition (ICPR)*, volume 2, Quebec City, August 2002.
16. L. J. Latecki and R. Lakämper. Convexity rule for shape decomposition based on discrete contour evolution. *Computer Vision and Image Understanding*, 73:441–454, 1999.
17. L. J. Latecki and R. Lakämper. Shape similarity measure based on correspondence of visual parts. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 22(10):1185–1190, 2000.
18. L. J. Latecki and R. Lakämper. Application of planar shapes comparison to object retrieval in image databases. *Pattern Recognition*, 35 (1):15–29, 2002.
19. L. J. Latecki and R. Lakämper. Polygon evolution by vertex deletion. In M. Nielsen, P. Johansen, O.F. Olsen, and J. Weickert, editors, *Scale-Space Theories in Computer Vision. Proc. of Int. Conf. on Scale-Space'99*, volume LNCS 1682, Corfu, Greece, September 1999.
20. L. J. Latecki, R. Lakämper, and U. Eckhardt. Shape descriptors for non-rigid shapes with a single closed contour. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, pages 424–429, South Carolina, June 2000.
21. Lu, F., Milios, E., Robot Pose Estimation in Unknown Environments by Matching 2D Range Scans. *Journal of Intelligent and Robotic Systems* 18:3 249–275, 1997.
22. F. Mokhtarian, S. Abbasi, and J. Kittler. Efficient and robust retrieval by shape content through curvature scale space. In A. W. M. Smeulders and R. Jain, editors, *Image Databases and Multi-Media Search*, pages 51–58. World Scientific Publishing, Singapore, 1997.

23. F. Mokhtarian and A. K. Mackworth. A theory of multiscale, curvature-based shape representation for planar curves. *IEEE Trans. PAMI*, 14:789–805, 1992.
24. Röfer, T., Using Histogram Correlation to Create Consistent Laser Scan Maps. *IEEE Int. Conf. on Robotics Systems (IROS)*. EPFL, Lausanne, Switzerland, 625–630, 2002.
25. K. Siddiqi, A. Shokoufandeh, S. J. Dickinson, and S. W. Zucker. Shock graphs and shape matching. *Int. J. of Computer Vision*, 35:13–32, 1999.
26. S. Thrun. Learning Metric-Topological Maps for Indoor Mobile Robot Navigation, *Artificial Intelligence* 99, pp. 21–71, 1998.
27. S. Thrun. Probabilistic algorithms in robotics. *AI Magazine*, 21(4):93–109, 2000.
28. S. Thrun. Robot Mapping: A Survey, In Lakemeyer, G. and Nebel, B. (eds.): *Exploring Artificial Intelligence in the New Millenium*, Morgan Kaufmann, 2002.
29. Thrun, S., Burgard, W., and Fox, D., A real-time algorithm for mobile robot mapping with applications to multi-robot and 3D mapping. *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2000.