Contour Grouping Based on Local Symmetry

Nagesh Adluru
Temple University, Philadelphia
nagesh@temple.edu

Longin Jan Latecki
Temple University, Philadelphia
latecki@temple.edu

Rolf Lakaemper
Temple University, Philadelphia
lakamper@temple.edu

Thomas Young
Temple University, Philadelphia
tdyoung@temple.edu

Xiang Bai
Huazhong Univ. of Sci. and Tech., Wuhan
xiang.bai@gmail.com

Ari Gross
Queens College, CUNY
ari@vision.cs.qc.edu

Abstract

The paper deals with grouping of edges to contours of shapes using only local symmetry and continuity. Shape skeletons are used to generate the search space for a version of the Markov Chain Monte Carlo approach utilizing particle filters to find the most likely skeleton. Intuitively this means that grouping of edge segments is performed by walking along the skeleton. The particle search, which is an adapted version of a successful algorithm in robot mapping, is assisted by a reference model of a shape, which is expressed as the sequence of sample points and radii of maximal skeleton disks. This model is sufficiently flexible to represent non-rigid deformations, but restrictive enough to perform well on real, noisy image data. The order of skeleton points (and their corresponding segments) found by the particles defines the grouping.

1. Introduction

The goal of the algorithm presented in this paper is to group pixels from an edge map of some digital image to a contour of a given target object. We interpret the problem of contour grouping as a SLAM (Simultaneous Localization and Mapping) problem as it is stated in the field of robot mapping. Recently breakthrough solutions to the SLAM problem have been obtained using particle filters [19, 7, 5]. Our algorithm adapts these approaches, which iterate over the processes of localization of the robot in the existing partial map, followed by a map update based on new observations. The initial guess of a new robot location is obtained from its previous location and from odometry readings. Since these odometry readings may be noisy, it is then corrected using new observations (usually obtained with a laser range finder). After this correction, the new observations are added to extend the existing partial map.

Our adaption treats the contour as the map that needs to be constructed. However since we do not have any real robot walking, the robot path is not explicitly given in our framework as is in the case of SLAM and hence do not have any odometry information. Therefore, further constraints on the robot path are needed in our framework. We set these constraints by assuming that our (virtual) robot is walking along the medial axis (MA). At every point of the MA (that is not endpoint or a junction point) the traversal direction is uniquely determined, and each MA point is equidistant to at least two different boundary points. Thus, we replace the odometry information with the constraint on the direction of traversal, which is obtained from a reference model. We replace the robot observations with the local symmetry constraint, which allows our robot to perceive only pairs of locally symmetric contour segments, and with a contour continuity constraint, which can be viewed as practical realization of Gestalt grouping principles [20].

The local symmetry constraint is based on the fundamental property of the MA that dates back to Blum [3]. According to Blum's definition, the medial axis, also called a skeleton S of a set D, is the locus of the centers of maximal disks. A maximal disk in D is a closed disk contained in D that is interiorly tangent to the boundary of D and that is not contained in any other disk in D. Each maximal disc must be tangent to the boundary in at least two different points. With every skeleton point $s \in S$ we also store the radius $r(s)$ of its maximal disk. An important property is that the skeleton can be computed for every planar shape. In order to generate a shape model, we first compute a skeleton of a
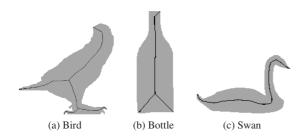
(a) Bird   (b) Bottle   (c) Swan

Figure 1: We use skeletons as our shape models.

given 2D shape with the method described in Bai et al. [1]. We use this method, since it produces very stable skeletons with only relevant branches. Another important property of this method is that all skeleton endpoints lie on the object contour. For example, three skeletons computed by this method are shown in Fig. 1.

Our shape model is a set of skeleton paths. For example, the shape model of the bottle is composed of two paths ending in the head. Each path is represented as a sequence of N sample points and the radii of maximal disks at the sample points. As stated above, we use a single model path as our reference model that replaces odometry for our virtual robot. Therefore, to group contours based on models with more than one path, we perform the proposed algorithm for each path separately. However, if one path is found, we know the approximate starting positions for the other paths in the image.

The processing flow of the proposed approach is as follows. For a given input image, we compute a gray level edge image. Then we group edge pixels to linear structures by applying an extended EM (Expectation Maximization) algorithm [9]. The obtained line segments are the input to the proposed contour grouping algorithm.

2. Related work

There is a huge number of papers on contour grouping that are based on various optimization principles. Therefore, we only review papers related to symmetry and particle filters. We also stress that the problem of contour grouping still remains an open problem.

The use of symmetry as a key contour grouping cue has been studied in both human vision and computer vision. Among others, the results in [13, 16, 10, 21] show that symmetry is non-accidental and therefore, can be used to distinguish salient contour structures from noisy background. Symmetry analysis of a given object boundary is usually conducted by computing its symmetry axis.

Symmetry principle expressed as global contour symmetry has been used in contour grouping in various approaches. One of the most recent approach that is based on global contour symmetry is presented in [18]. It is related to the proposed work is the grouping method developed by Mohan and Nevatia [14], where symmetry is considered along with closure and proximity. Symmetry is applied as a cue to pair the extracted curves by producing a set of ribbons. These ribbons are then grouped into some structures by some heuristic algorithms.

Our work that is related to the grouping method developed by Liu, Geiger, and Yuille [12] in that local symmetry axes are used. They identify the local symmetry-axis segments and then apply a shortest-path algorithm to connect some of them into a complete symmetry axis. The grouping cost function is defined as the sum of local costs along the symmetry axis. In addition to using different measures, we have a more powerful computational framework in the proposed approach. Integer Quadratic Programming is is used in [17] to group contour segments based on constrained Delaunay triangulation, where line segments were first fitted to edge images. The grouping is performed with a reference model that resembles a skeleton composed of line segments representing human body parts.

Particle filters (also called sequential Monte Carlo methods) have extensively been applied for robust object tracking. One of the best-known approaches is the Condensation algorithm [8], which allows tracking object contours the presence of background clutter. A sophisticated version of particle filter algorithm was applied to contour detection in [15]. The detection is performed on edge pixels with particles following the contour directly. In this paper also a simple ribbon geometry is used for road extraction.

Particle filtering has become the standard approach for mobile-robot localization with the main application being SLAM [19, 7, 5], where probability distributions for the robot poses (position plus heading direction) and the possible maps are approximated and propagated by a set of particles. As stated in the introduction, our approach is closely related to the SLAM, since we cast contour grouping as map building, but the geometric interpretation is based on local symmetry of contours with respect to their medial axes.

In comparison to previous approaches our work has at least two serious advantages. We are able to group contour segments in the presence of distractor segments between local symmetric contour pieces. The grouping under such conditions seems to be impossible in [12, 17, 15]. Even if our shape models are derived from complete contours, grouping of only parts of contours is possible in our framework. This is in contrast with active contours based methods [2].

3. Contour grouping with particle filtering

Our approach is inspired by the robot mapping approach in Grisetti et al. [7]. The key idea of the Rao-Blackwellized particle filter for SLAM is to estimate a pos-

terior $p(x_{1:t}|z_{1:t}, u_{1:t})$ over potential trajectories $x_{1:t}$ of the robot given its observations $z_{1:t}$ and its odometry measurements $u_{0:t}$ and to use this posterior to compute a posterior over maps m and trajectories:

$$p(x_{1:t}, m_t|z_{1:t}, u_{1:t}) = p(m_t|x_{1:t}, z_{1:t})p(x_{1:t}|z_{1:t}, u_{1:t}) \tag{1}$$

In our context, the robot trajectory $x_{1:t}$ is a skeleton path and the contour is the map m_t that we need to construct. We replace odometry measurements $u_{1:t}$ with the comparison to a reference model. Our reference model is simply a sequence of MA points and the radii of maximal disks along a skeleton path in the reference shape. We do not have any error in our odometry, since our virtual robot moves precisely, but our "odometry" error comes from the reference model itself, since we weaken the constraints on the model accuracy. By this the reference model becomes very flexible, since we allow for large inaccuracy in the position of MA points. Finally, an observation z_t at the trajectory point x_t comprises from the estimated radius of maximal disk with respect to two contour segments and the two tangential points of the maximal disk to the two contour segments. Thus, our observation at a given trajectory point x_t is based on two contour pieces that are locally symmetric at x_t. Since not every point in a given edge image is a center of some maximal disk, the robot trajectories are restricted to a subset of a given digital image.

In our context the posterior over maps $p(m_t|x_{1:t}, z_{1:t})$ is defined based on contour smoothness, which can be viewed as a practical realization of the good continuation rule, and the local symmetry measured with respect to the radii of maximal disks along a skeleton path in the model. Observe that we evaluate contour smoothness from a view of centers of maximal disks, which gives us a better view than viewing the contour from other contour points (which is usually the case): since we walk along the skeleton we are also able to use local symmetry to evaluate the contour smoothness. The posterior over the potential trajectories $p(x_{1:t}|z_{1:t}, u_{1:t})$ is evaluated based on the predicted position with respect to the reference model.

Although the interpretation of robot mapping terms and the posterior evaluations are different, our computation framework is similar to that in SLAM. To estimate the posterior $p(x_{1:t}|z_{1:t}, u_{1:t})$ over the potential trajectories, an individual map is associated with every sample. In analogy, an individual grouping of edge line segments (that is expected to form the contour) is associated with every sample. Contour is defined here as a sequence of locally symmetric line segments. Each contour is built given the observations $z_{1:t}$ and the trajectory $x_{1:t}$ represented by a corresponding particle.

Following the approach in [7], we employ one of the most common particle filtering algorithms: the Rao-Blackwellized Sampling Importance Resampling (SIR) fil-

ter. It incrementally processes the observations and the readings from the reference model, which is done by updating a set of samples representing the posterior about the contour and the trajectory of our virtual robot. The following four steps are executed:

1) **Sampling:** The next generation of particles $\{x_t^{(i)}\}$ is obtained from the current generation $\{x_{t-1}^{(i)}\}$ by sampling from a proposal distribution $\pi(x_{1:t}|z_{1:t}, u_{1:t})$ which is assumed to satisfy the following recursion based on 1st on first order Markov assumption:

$$\pi(x_{1:t}|z_{1:t}, u_{1:t}) = \pi(x_t|x_{1:t-1}, z_{1:t}, u_{1:t}) \cdot \\ \pi(x_{1:t-1}|z_{1:t-1}, u_{1:t-1}) \tag{2}$$

2) **Importance Weighting:** An individual importance weight $w^{(i)}$ is assigned to each particle, according to

$$w^{(i)} = \frac{p(x_{1:t}|z_{1:t}, u_{1:t})}{\pi(x_{1:t}|z_{1:t}, u_{1:t})} \tag{3}$$

The weights $w^{(i)}$ account for the fact that the proposal distribution π in general is not equal to the true distribution of successor states.

3) **Resampling:** Particles with a low importance weight w are typically replaced by samples with a high weight. This step is necessary since only a finite number of particles are used to approximate a continuous distribution. Furthermore, resampling allows application of a particle filter in situations in which the true distribution differs from the proposal.

4) **Contour Estimating:** For each pose sample $x_t^{(i)}$, the corresponding contour estimate $m_t^{(i)}$ is computed based on the trajectory and the history of observations according to $p(m_t^{(i)}|x_{1:t}^{(i)}, z_{1:t})$.

Our approach benefits from two improvements described in [7]. The first is the computation of the optimal proposal distribution, which has been successfully applied in FastSLAM-2 [12] for landmark-based mapping. The second is adaptive resampling technique, which was introduced in [7] for Rao-Blackwellized particle filters.

As described in the prediction step (1), we need to draw samples from a proposal distribution $\pi(x_t|x_{1:t-1}, z_{1:t}, u_{1:t})$. Following Doucet [4] the optimal choice of the proposal distribution with respect to the variance of the particle weights and under the Markov assumption is given as

$$p(x_t|m_{t-1}^{(i)}, x_{t-1}^{(i)}, z_t, u_t) = \frac{p(z_t|m_{t-1}^{(i)}, x_t)p(x_t|x_{t-1}^{(i)}, u_t)}{\int p(z_t|m_{t-1}^{(i)}, x)p(x|x_{t-1}^{(i)}, u_t)dx} \tag{4}$$

With this optimal proposal the importance weights can be recursive computed as:

$$w_t^{(i)} = w_{t-1}^{(i)} \int_x p(z_t|m_{t-1}^{(i)}, x)p(x|x_{t-1}^{(i)}, u_t)dx \tag{5}$$

In the case of a robot equipped with the laser range finder, the likelihood function of the odometry motion model $p(x_t|x_{t-1}^{(i)}, u_t)$ is very flat while $p(z_t|m_{t-1}^{(i)}, x_t)$ based on laser readings is extremely peaked. The situation is similar in our application, therefore, we follow the solution introduced in [7]. Since the product $p(z_t|m_{t-1}^{(i)}, x_t)p(x_t|x_{t-1}^{(i)}, u_t)$ is dominated by $p(z_t|m_{t-1}^{(i)}, x_t)$, we can approximate the integral in Eq. (5) as:

$$w_t^{(i)} \approx w_{t-1}^{(i)} \sum_{j=1}^{K} p(z_t|m_{t-1}^{(i)}, x_j) \qquad (6)$$

where $x_j \in \{x_t|p(x_t|x_{t-1}, u_t) > \alpha\}$, where α is a threshold related to the accuracy of the reference model. Moreover, since obtaining a closed form for the optimal proposal (Eq. (4)) is very hard it is approximated by a Gaussian distribution.

$$p(x_t|m_{t-1}^{(i)}, x_{t-1}^{(i)}, z_t, u_t) \approx \phi(\mu_t^{(i)}, \Sigma_t^{(i)}). \qquad (7)$$

The Gaussian parameters, $\mu_t^{(i)}, \Sigma_t^{(i)}$, are computed using the K samples as:

$$\mu_t^{(i)} = \frac{1}{\nu^{(i)}} \sum_{j=1}^{K} x_j p(z_t|m_{t-1}^{(i)}, x_j) \qquad (8)$$

$$\Sigma_t^{(i)} = \frac{1}{\nu^{(i)}} \sum_{j=1}^{K} (x_j - \mu_t^{(i)})^T (x_j - \mu_t^{(i)}) p(z_t|m_{t-1}^{(i)}, x_j) \qquad (9)$$

where $\nu^{(i)} = \sum_{j=1}^{K} p(z_t|m_{t-1}^{(i)}, x_j)$.

Consequently, in each update of the particle filter, we only need to compute $p(z_t|m_{t-1}^{(i)}, x_j)$ and $p(x_t|x_{t-1}^{(i)}, u_t)$. Both probabilities are directly computed using our geometric model.

We also use selective resampling in our approach. During resampling, particles with a low importance weight are usually replaced by samples with a high weight. Resampling is necessary since only a finite number of particles is used. However, after resampling several times only very few particles might be duplicated and good samples might be deleted from the sample set, causing "particle depletion" problem. To reduce this effect in selective resampling, resampling operations are only performed when needed. As the criterion to decide when the resampling is needed we use the N_{eff} measure [11]:

$$N_{eff} = \frac{1}{\sum_{1=1}^{N} (w^{(i)})^2}, \qquad (10)$$

where N is the number of particles. It is based on the observation that if the samples were drawn from the true posterior, the importance weights of the samples would be equal to each other. The worse the approximation the higher the variance of the importance weights. Since N_{eff} can be regarded as a measure of the dispersion of the importance weights, it is a useful measure to evaluate how well the particle set approximates the true posterior. Following [7] and [4], we resample each time N_{eff} drops below a threshold of $N/2$.

4. Geometric construction of center points

For a given set of line segments (which approximate edge pixels), we describe a construction of all potential medial axis (MA) points. Since our construction only approximates MA points, we call the constructed points *center points*. Center points are candidates for the reconstructed skeleton points, i.e. the set of center points defines the search space for particles. In other words, each particle is a sequence of center points.

Center points are determined for pairs of line segments, their construction is motivated by the classical definition of skeleton points on closed curves [3] (centers of maximal inscribing disks). In contrast, our construction of center points is performed for a non-connected set of line segments. Since the line segments are obtained by fitting edge pixels, their directions do not precisely represent the directions of the true contours. Hence the task of computing the center points is non trivial. The following definitions refer to Fig. 2.
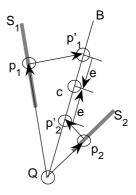

Figure 2: Construction of center points.

Let S_1, S_2 be two line segments and P_1, P_2 two sets of points on S_1 and S_2 respectively, gained from equidistant sampling of S_1, S_2 with a sample size ϵ. Then the set $C(S_1, S_2)$ of segment center points of S_1 and S_2 is the union $\cup c(p_1, p_2)$ of center points of all pairs of sample points (p_1, p_2) in $(P_1 \times P_2)$, which is defined as follows:

Let $p_1, p_2 \in (P_1 \times P_2)$, and d_1, d_2 be the direction vectors of S_1, S_2. Let Q be the intersection between the lines containing S_1, S_2, and B the bisecting half straight that halves the angle $\angle(Qp_1, Qp_2)$. Let p_1', p_2' be the projection points of p_1 and p_2 onto B, orthogonal to d_1 and d_2 respectively (not orthogonal to B). Then the center point $c(p_1, p_2)$ of points p_1, p_2 is defined as the mid point $(p_1' + p_2')/2$ between p_1' and p_2'. In the case of $S_1 \parallel S_2$, $c(p_1, p_2)$ is defined as $(p_1 + p_2)/2$, which is a continuous extension and in accordance with the construction.

In the case of sub sampling with a sample size of $\epsilon \to 0$ this definition is a generalization of the classical definition: if PC_1, PC_2 denotes a pair of points on a continuous polygon that define a maximal disk, then the above construction yields the center point of this disk (using the tangent directions of the polygon at PC_1, PC_2 as the directions d_1, d_2). Hence we generate a superset of the skeleton points if the set of sample points (here: on the polygon) contain PC_1 and PC_2. This is guaranteed if $\epsilon \to 0$. The distance e between p_1' and c (or p_2' and c, see Fig. 2) can be interpreted as a quality measure for $c(p_1, p_2)$ to be the center of a maximal disk. If $e = 0$ (best value), the segments S_1 and S_2 are tangents to the circle defined by the center c and the points p_1, p_2. The smaller e, the closer is c to a skeleton point in the classical sense. Since in real applications we use a sample size $\epsilon > 0$, most center points in C will gain a measure $e > 0$. To extract a set of center points that represents intuitive skeleton candidates, we allow e to be in $[0, T_e]$ with a certain threshold T_e. Point combinations exceeding T_e will be removed and not be used as skeleton candidates. The special cases of $T_e = 0$ or $\epsilon \to 0$ lead to the following relations between C and skeleton points:

1. $\epsilon \to 0, T_e > 0 \Rightarrow C \supset CS$;

2. $\epsilon \to 0, T_e = 0 \Leftrightarrow C = CS$;

3. $\epsilon > 0, T_e = 0 \Rightarrow C \subset CS$.

Fig. 3 shows a set C of generated skeleton candidates of two segments. The sample points on the two segments are shown in black, while the skeleton candidates (or center points) are shown in red. The dashed line shows the center points which are generated by the outer segments. Observe that the middle segment (which could be seen as noise if the outer segments are part of the true contour) generates additional skeleton candidates, which have to be disregarded during the particle filtering process.

5. Contour smoothness

This section describes the geometric construction of the contour smoothness measure $S(c_1, c_2)$, defined for pairs of center points (c_1, c_2). It describes the symmetry or property of good continuation of the transition from the seg-
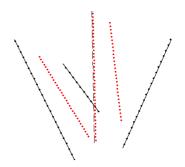

Figure 3: Skeleton candidates of three segments (solid lines). The sample points on the two segments are shown in black, while the skeleton candidates (or center points) are shown in red.

ments S_1, S_2 corresponding to center point c_1 to the segments S_3, S_4 corresponding to center point c_2. All definitions in this section refer to Fig. 4.
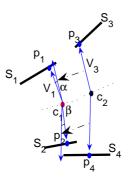

Figure 4: Evaluation of contour smoothness.

The smoothness measure is composed of two parts, taking into account the distance and the angle of continuation. First we define the distance compound.

Let $c_1 = c(p_1, p_2)$ be the center point induced by points $p_1 \in S_1, p_2 \in S_2$. We call the pair (S_1, S_2) the source segments of c_1. Let (S_3, S_4) be the source segments of the second center point c_2. We define the source segment distance $D_S(c_1, c_2)$ as the minimal sum of distances of possible continuations of S_1, S_2 on S_3, S_4:

$$D_S(c_1, c_2) = \tag{11}$$
$$\min(d_S(S_1, S_3) + d_S(S_2, S_4), d_S(S_1, S_4) + d_S(S_2, S_3))$$

with the distance $d_S(S_i, S_j)$ between two line segments S_i, S_j defined below. D_S defines a distance value as well as the pairing of the segments, i.e. the configuration of continuation. For example, in Fig. 4, S_1 continues S_3, S_2 con-

tinues S_4, since $d_S(S_1, S_3) + d_S(S_2, S_4) < d_S(S_1, S_4) + d_S(S_2, S_3)$).

Let S_i, S_j be two line segments with endpoints $e1_i, e2_i$ and $e1_j, e2_j$. The distance $d_e(p_i, S_j)$ of a point $p_i \in S_i$ to a segment S_j is defined as the minimal Euclidean distance between p_i and all points $p_j \in S_j$. Then the distance between two line segments is defined as

$$d_S(S_1, S_2) = \qquad\qquad\qquad (12)$$
$$\min(d_e(e1_i, S_j), d_e(e2_i, S_j), d_e(e1_j, S_i), d_e(e2_j, S_i))$$

The second compound measures the angular distance based on the segment configuration established by D_S. With every center point c_1, we have two vectors associated that emanate from c_1 to its two line segments S_1, S_2. Let $p_1 \in S_1, p_2 \in S_2$ be the points that construct the center point $c(p_1, p_2)$. We call p_1, p_2 the source points of $c(p_1, p_2) = c_1$. We define $V_i = p_i - c(p_i, p_j)$ and $V_j = p_j - c(p_i, p_j)$, (e.g. Fig. 4, $V_3 = p_3 - c(p_3, p_4)$). Then the angular distance between two center points is defined as $A_S(c_1, c_2) = \angle(V_1, V_2) + \angle(V_3, V_4)$. For example, in Fig. 4, $A(c_1, c_2)$ is the sum of the angles $\alpha + \beta$. Finally, the smoothness measure $C(c_1, c_2)$ is defined as mixture of two Gaussians:

$$C(c_1, c_2) = \eta \, \phi_{\sigma_1}(D_S(c_1, c_2)) + (1 - \eta) \, \phi_{\sigma_2}(A_S(c_1, c_2))$$
$$(13)$$

with η steering the influence of the distance and angular terms. In all our experiments, we set $\eta = 0.5$.

6. Particle evaluation

In this section, we describe the computation of $p(z_t | m_{t-1}^{(i)}, x_j)$ needed for particle weight computation in formula ((5)). The particle $x_{t-1}^{(i)}$ is represented as a sequence of $t - 1$ center points, and their associated radii of maximal disks. Let c_1 be the last center point in this sequence. $m_{t-1}^{(i)}$ represents the line segments grouped so far by $x_{t-1}^{(i)}$. These are the pairs of line segments that generated the center points. In our framework, the observation z_t represents the expected radius R_t of a maximal disk that we retrieve from the reference model. We consider x_j as a center point c_2 that is a possible continuation of $x_{t-1}^{(i)}$. Let R_j be the radius of the maximal disk at the center point c_2. We define the model fit with a single Gaussian

$$M(c_2) = \phi_{\sigma_3}(R_j - R_t), \qquad\qquad (14)$$

where σ_3 defines the tolerance related to the model accuracy. Finally, we obtain

$$p(z_t | m_{t-1}^{(i)}, x_j) = C(c_1, c_2) M(c_2) \qquad (15)$$

Thus, each particle is judged for fitness by two criteria. The first is the contour smoothness, the second the reference

model fitness. The two distributions C and M are independent, since the probability of a certain sequence of radii is not related to smoothness in a given gray level edge image.

It remains to define the distribution $p(x_t | x_{t-1}^{(i)}, u_t)$. As stated in Section 3, we replace the inaccuracy in the odometry readings u_t with the imprecision of our shape model. In our context, $p(x_t | x_{t-1}^{(i)}, u_t)$ is simply a 3D Gaussian distribution over x and y coordinates of the location of sample points on a model skeleton path and the radii at the sample points. Here we have a clear tradeoff between the shape generality of our model and its precision. The more precise our model is, the less general is the shape class represented by the model. It is possible to learn the model from a given class of shapes, by first aligning the shapes, and then computing the distributions at given sample points. We simply constructed our models by computing skeletons of single binary shapes [1], and setting the model variance manually.

7. Experimental results

We provide experimental results illustrating the proposed contour grouping. We used images form the ETHZ database [6], since it contains a gray level edge map for each image, which allows for a fair performance comparison. It also contains model object, which we used to generate our shape reference models. The reference models we used are the skeleton paths shown in Fig. 1. Each path was represented as a sequence of 50 equidistant sample points and the radii of maximal disks centered at the sample points.

Fig. 5 illustrates several contour grouping examples computed with the proposed method. The grouped contour line segments are shown in red. Their connections to corresponding center points are also shown in red. We also see the computed skeleton paths as blue circles, the set of all center points, and all the input edge segments. The original input images and the input gray level edge maps are shown in Fig. 6. The first two examples show grouping of two different objects in the same image (Fig. 5(a,b)). It is possible in our framework, since two sets of particles follow two different reference models. In all our experiments we used 100 particles for each shape reference model and 50-100 time steps.

Fig. 7 shows the particles as filtering progresses for the swan in Fig. 5(e). Each trajectory (particle) is shown in different color. We can observe that all through the process sufficiently diverse particles (hypotheses) are maintained while "bad" particles disappear towards the end as the filter converges.

8. Conclusions

We map the problem of contour grouping to a SLAM problem as it is stated in the field of robot mapping. We extend the particle filters based approach to SLAM so that
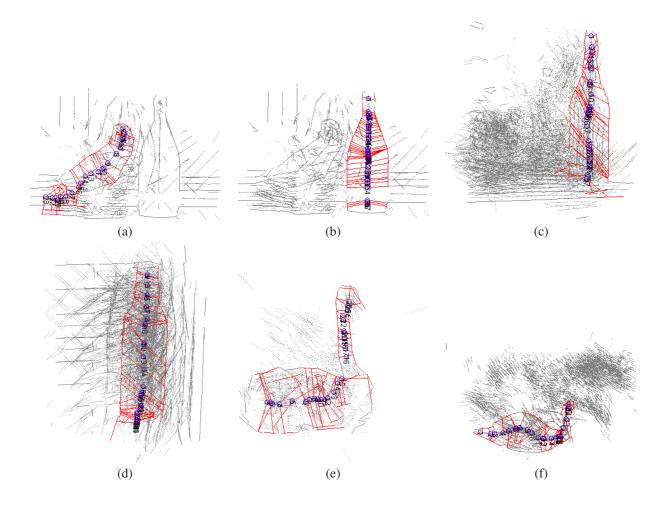
Figure 5: The grouped contour line segments are shown in red. Their connections to corresponding center points are also shown in red. We also see the computed skeleton paths as blue circles, the set of all center points, and all the input edge segments. The original images are shown in Fig. 6.

Figure 6: The input images and the input edge maps for grouped contours shown in Fig. 5.

statistical inference based on a reference model is possible. In comparison to previous approaches our work has at least two serious advantages that are demonstrated in our exper-

imental results. We are able to group contour segments in the presence of distractor segments between local symmetric contour pieces. Even if we our shape models are derived
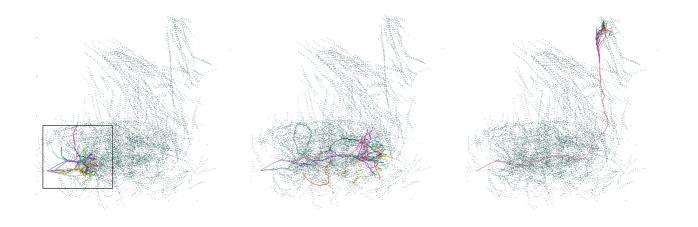
Figure 7: The evolution of particles at iterations 16, 61, and 96. Only center points and the trajectories of particles are shown for clarity of presenting.

from complete contours, grouping of only parts of contours is possible.

Acknowledgment

References

[1] X. Bai, L. J. Latecki, and W. Y. Liu. Skeleton pruning by contour partitioning with discrete curve evolution. *IEEE Trans. on PAMI*, 29(3):449–462, 2007. 2, 6

[2] A. Blake and M. Isard. *Active Contours*. Springer-Verlag, 1997. 2

[3] H. Blum. A transformation for extracting new descriptors of shape. In W. Wathen-Dunn, editor, *Models for the Perception of Speech and Visual Form*. MIT Press, Cambridge, 1967. 1, 4

[4] A. Doucet. *On sequential simulation-based methods for bayesian filtering*. Technical report, Signal Processing Group, Dept. of Engeneering, University of Cambridge, 1998. 3, 4

[5] A. Eliazar and R. Parr. DP-SLAM: Fast, robust simulataneous localization and mapping without predetermined landmarks. In *Int. Joint Conf. on Artificial Intelligence (IJCAI)*, 2003. 1, 2

[6] V. Ferrari, T. Tuytelaars, and L. J. V. Gool. Object detection by contour segment networks. In *ECCV*, pages 14–28, 2006. 6

[7] G. Grisetti, C. Stachniss, and W. Burgard. Improved techniques for grid mapping with rao-blackwellized particle filters. *IEEE Trans. on Robotics*, 23:34–46, 2007. 1, 2, 3, 4

[8] M. Isard and A. Blake. Condensation – conditional density propagation for visual tracking. *Int. Journal of Computer Vision*, 29(1):5–28, 1998. 2

[9] L. J. Latecki, M. Sobel, and R. Lakaemper. New EM derived from Kullback-Leibler divergence. In *ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining (KDD)*, 2006. 2

[10] M. Leyton. *Symmetry, Causality, Mind*. MIT Press, Cambridge, 1992. 2

[11] J. S. Liu. Metropolized independent sampling with comparisons to rejection sampling and importance sampling. *Statist. Comput.*, 6:113–119, 1996. 4

[12] T. Liu, D. Geiger, and A. L. Yuille. Segmenting by seeking the symmetry axis. In *Proc. CVPR*, pages 994–998, 1998. 2

[13] D. G. Lowe. *Perceptual Organization and Visual Recognition*. Kluwer Academic Publishers, Boston, 1985. 2

[14] R. Mohan and R. Nevatia. Perceptual organization for scene segmentation and description. *IEEE Trans. on PAMI*, 14(6):616–635, 1992. 2

[15] P. Perez, A. Blake, and M. Gangnet. Rjetstream: Probabilistic contour extraction with particles. In *Proc. ICCV*, pages 524–531, 2001. 2

[16] A. P.Witkin and J. M. Tenenbaum. On the role of structure in vision. In J. Beck, B. Hope, and A. Rosenfeld, editors, *Human and Machine Vision*. Academic Press, New York, 1983. 2

[17] X. Ren, A. Berg, and J. Malik. Recovering human body configurations using pairwise constraints between parts. In *Proc. ICCV*, 2005. 2

[18] J. S. Stahl and S. Wang. Globally optimal grouping for symmetric boundaries. In *Proc. CVPR*, 2006. 2

[19] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. The MIT Press Cambridge, 2005. 1, 2

[20] M. Wertheimer. Untersuchungen zur lehre von der gestalt ii. *Psycologische Forschung*, 4:301–350, 1923. 1

[21] S. C. Zhu and A. Yuille. Forms: a flexible object recognition and modelling system. In *Proc. ICCV*, pages 465–472, 1995. 2