Shape Guided Contour Fragment Grouping with Particle Filters

ChengEn Lu
Huazhong University of Sci. and Tech.
Wuhan, 430074, China
luchengen@gmail.com

Longin Jan Latecki
Temple University
Philadelphia, 19122, USA
latecki@temple.edu

Nagesh Adluru
University of Wisconsin-Madison
Madison, 53705, USA
adluru@wisc.edu

Haibin Ling
Temple University
Philadelphia, 19122, USA
hbling@temple.edu

Xingwei Yang
Temple University
Philadelphia, 19122, USA
xingwei@temple.edu

Abstract

We propose a novel framework for contour based object detection and recognition, which we formulate as a joint contour fragment grouping and labeling problem. For a given set of contours of model shapes, we simultaneously perform selection of relevant contour fragments in edge images, grouping of the selected contour fragments, and their matching to the model contours. The inference in all these steps is performed using particle filters (PF) but with static observations. Our approach needs one example shape per class as training data. The PF framework combined with decomposition of model contour fragments to part bundles allows us to implement an intuitive search strategy for the target contour in a clutter of edge fragments. First a rough sketch of the model shape is identified, followed by fine tuning of shape details. We show that this framework yields not only accurate object detections but also localizations in real cluttered images.

1. Introduction

The key role of contours and their shapes in object extraction and recognition in images is well established in computer vision and in visual perception. Extracting edges in digital images is relatively well-understood and there are robust detectors like [20, 11]. However, it is often difficult to distinguish edge pixels corresponding to meaningful object contours. The main problem is that usually most edge pixels represent background and irrelevant texture, i.e., clutter, and only a small subset of edge pixels corresponds to object contours. Further, the edge pixels do not simply form occluding contours but broken contour fragments due to noise and occlusion. Thus, the target occluding contour

may have large gaps in local edge detection, which renders any (bottom-up) local search for occluding contours in cluttered images unsuccessful. The occluding contours cannot be simply extracted by template matching, since the shape of objects in images varies significantly due to view point change, non rigid deformation, and occlusion. Clutter combined with contour gaps makes contour grouping a very difficult task that requires global (top-down) information. A further difficulty stems from the fact that contours of target objects form only a small portion of edge pixels, which is often less than two or three percent. Thus, we deal with an unusually high noise to signal ratio. We show a few example edge images illustrating these problems in Fig. 1. Interestingly, human visual system still can perform contour grouping, object detection, and recognition, even though only cluttered contour information is provided (there is no texture and color in these images). Humans can easily perform all these tasks although important contour information is missing, and we may not be able to complete the missing contour parts, e.g., we can recognize a giraffe, but we may not be able to draw or imagine the missing outline of the giraffe's head. Thus, humans can perform contour grouping, object detection, and recognition while keeping at least part of missing information ambiguous, and we do not attempt to disambiguate all missing information.
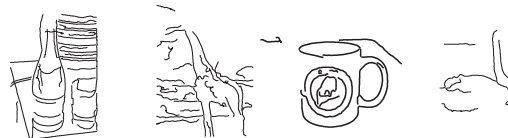

Figure 1. Parts of the objects are missing both due to missing edges and due to broken edge links.

As illustrated in Fig. 1 shape information alone is often sufficient for boundary detection. Since shape information is invariant to color, texture, and brightness, we can sig-

nificantly reduce the number of training examples needed for object recognition. In fact, in the proposed approach we only use a small number of hand drawn occluding contours as our shape models. Given an input image, e.g., Fig. 2(a), we perform edge detection (b), and group edge pixels to edge fragments in a simple bottom-up process (c). We detect a global geometric configuration of edge fragments that is most similar to a given shape model by simultaneously performing top-down selection of foreground edge fragments and shape matching process. Since our target function is highly discontinuous, and exhaustive brute force search of all possible global configurations of edge fragments has a prohibitive complexity, we employ a particle filter (PF) framework to drive the top-down computation. We extend the standard PF framework to address the following two issues that are specific to our task: (1) Since all edge fragments in a given image are available, we have a case of PF with static observations. (2) In our approach particles perform "tracking" in the space of pairs of model contour fragments and edge fragments in the image. Thus, in our application time corresponds to the number of pairs in this space, which also corresponds to the number of grouped edge fragments in the image. Fig. 2(d) shows a detected and recognized swan. It is composed of four edge fragments, which means that our PF algorithm required four steps. The model swan used is shown in the top left corner.
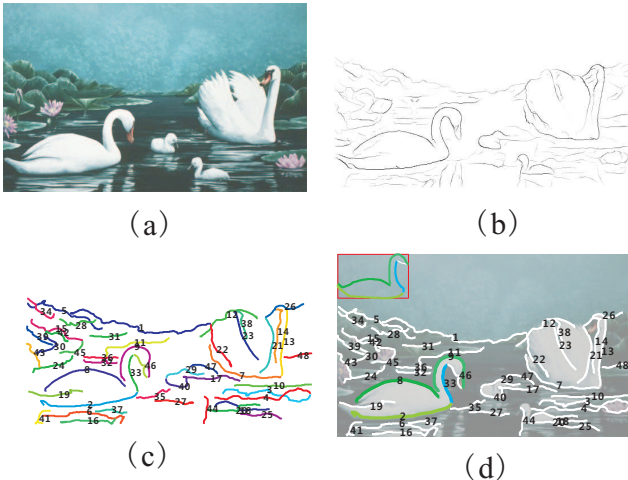

Figure 2. (b) show the edge map of image (a). (c) The 48 edge segments obtained by simple edge linking form our label set $E = \{e_1, e_2, \ldots e_{48}\}$. (d) The grouping result obtained by the proposed PF approach. The colors indicate the assignment to the model segments shown in top left.

The main problem in recognizing objects of interest using edge fragments obtained from cluttered images seems to be the fact that a sufficient number of contour fragments needs to be assembled in order to make any useful detection decision. For example, any two contour fragments of the detected swam in Fig. 2(d) are not sufficiently discrimina-

tive. To address this problem a "delayed decision" strategy is necessary, and PF provides a statistically sound approach to achieve delayed decision. Moreover, PF allows us to realize an intuitive idea of first assembling a rough sketch of the target contour from edge fragments, and then refining this sketch by adding further edge fragments if the shape similarity to the target contour can be improved.

The rest of the paper is organized as follows. After briefly reviewing related work in §2, we present the core theory of our system in §3. Our flexible part bundle shape model is described in §4. It constrains the proposal distribution of PF to first follow a rough sketch of the target contour. We describe our new shape descriptor in §5. Its flexibility makes it very suitable for contour fragment grouping. Finally, §6 presents our experimental results.

2. Related Work

Grouping edge pixels into contours using various saliency measures and cues has been studied for long and is still an active research field [33, 31, 27, 16, 26]. Once contours are identified they can be further grouped into objects by performing shape matching with model contours. For example, Shotton *et al*. [25] and Opelt *et al*. [22] use Chamfer distance [2] to match fragments of contours learnt from training images to edge images. McNeil and Vijayakumar [21] represent parts learnt from semi-supervised training as point-sets and establishes probabilistic point correspondences for the points in edge images. Ferrari *et al*. [7] use a network of nearly straight contour fragments and sliding window search. Thayananthan *et al*. [28] modify shape context [1] to incorporate edge orientations and Viterbi optimization for better matching in clutter. Flezenswalb and Schwartz [6] presents a shape-tree based elastic matching among two shapes and extended it to match a model and cluttered image by identifying contour parts (smooth curves) using [5]. More recently, Zhu *et al*. [34] formulate the shape matching of contours (identified using [33]) in clutter as a set-set matching problem. They present an approximate solution to the hard combinatorial problem by using a voting scheme ([32, 18]) and a relaxed context selection scheme by algebraically encoding shape context into linear programming. Besides, Ravishankar *et al*. [23] introduced a multi-stage contour based detection approach. They decompose the model shapes into segments at high curvature points. The segments are then scaled, rotated, deformed and matched independently in the gradient image. Dynamic programming is used to group the matched segments in a multi-stage process that begins with triples of segments as opposed to pairs of segments used in [7]. An appearance based approach was recently used by Maji and Malik [19] by integrating hough transform based features of codebooks into kernel classifiers. Aside from the recent work mentioned above, there are many early studies that

use geometric constraints for model-based object and shape matching [15, 13, 14].

3. Particle Filter with Static Observations

Let $S = \{s_1, \ldots, s_m\}$ be a set of model contour parts (or segments), which are called sites in the Markov Random Field (MRF) terminology. Let $E = \{e_1, \ldots, e_n\}$ be a set of edge fragments in a given image I, which are called labels in the MRF terminology. Edge fragments are generated using edge-linking software [17], then we introduce break points at high curvature points in order to allow for more flexible shape matching. Let E^S be a set of all functions $f : S \to E$, which represent label assignments to the sides. Our goal is to find a function f with maximum posterior probability for a pdf $p : E^S \to \mathbb{R}_+$:

$$\widehat{f} = \operatorname*{argmax}_{f \in E^S} p(f \mid Z), \tag{1}$$

where $\mathbb{R}_+$ denotes the nonnegative real numbers and Z is the set of the observations. It is a very important property of our framework that the set of observations Z is static. It is determined by the appearance of a target object (or a set of target objects). Our primary appearance feature is shape of the contour of the target object (or target objects). It is measured by shape similarity of extracted and grouped edge fragments to the target contour. The appearance features of target objects (which we also call model objects) can be manually defined, e.g., by drawing the contour of the query shape, or learned from training examples.

We propose to perform the optimization in Eq. 1 in the particle filter (PF) framework. This is possible, since each function $f \in X^S$ is a finite set of pairs, i.e., $f = \{x_1, \ldots, x_m\}$, where $x_k \in S \times E$. Obviously the order of the x's does not matter, i.e., each permutation of x's in the set $f = \{x_1, \ldots, x_m\}$ defines the same function f. A key observation for the proposed approach is that a sequence $(x_1, \ldots, x_m)$ maximizing p clearly determines the set $f = \{x_1, \ldots, x_m\}$ that maximizes p[1]. Thus, we solve a more specific problem of finding a sequence. The sequence order is important in the proposed PF inference process, which is described below. Following a common notation in the PF framework, we denote a sequence $(x_1, \ldots, x_m)$ as $x_{1:m}$. We can now restate our goal (1) as finding a sequence with maximum posterior probability:

$$\widehat{x}_{1:m} = \operatorname*{argmax}_{x_{1:m} \in (S \times E)^m} p(x_{1:m} \mid Z). \tag{2}$$

Obviously, as stated above, a solution $\widehat{x}_{1:m} = (\widehat{x}_1, \ldots, \widehat{x}_m)$ of Eq. 2, which is a sequence, defines a solution of Eq. 1 as $\widehat{f} = \{\widehat{x}_1, \ldots, \widehat{x}_m\}$, which is a set.

---

[1]The issue of initializing the sequence is discussed later in the section.

We will approximate $p(x_{1:m} \mid Z)$ in Eq. 2 in the framework of Bayesian Importance Sampling. By drawing samples $x_{1:m}^{(i)}$ for $i = 1, \ldots, N$ from an easier to sample proposal distribution π we obtain:

$$\widehat{p}(x_{1:m} \mid Z) = \sum_{i=1}^{N} w(x_{1:m}^{(i)}) \, \delta(x_{1:m} - x_{1:m}^{(i)}), \tag{3}$$

where δ is the Dirac delta function and

$$w(x_{1:m}^{(i)}) = \frac{p(x_{1:m}^{(i)} \mid Z)}{\pi(x_{1:m}^{(i)} \mid Z)} \tag{4}$$

are normalized weights. The weights $w(x_{1:m}^{(i)})$ account for the fact that the proposal distribution π in general is not equal to the true distribution of successor states.

However, due to the high dimensionality of $(S \times E)^m$ it is still hard to sample from π. Therefore, we will derive a recursive estimation of the weights and recursive sampling of the sequence elements one by one from $S \times E$. The recursive estimate of the importance weights will be obtained by factorizing the distributions p and π and by modeling the evolution of the hidden states $x_k \in S \times E$ in discrete time steps. We do not have any natural time parameter in our approach, but discrete steps of expanding the sequence of hidden states by one new state can be interpreted as discrete time steps. Our derivation is similar to the PF derivation, but it differs fundamentally, since unlike the standard PF framework, the observations Z do not arrive sequentially, but are available at once. For every t from 1 to m, we have

$$
\begin{aligned}
w(x_{1:t}^{(i)}) &= \frac{p(x_{1:t}^{(i)} \mid Z)}{\pi(x_{1:t}^{(i)} \mid Z)} = \frac{p(x_t \mid x_{1:t-1}^{(i)}, Z) \, p(x_{1:t-1}^{(i)} \mid Z)}{\pi(x_t \mid x_{1:t-1}^{(i)}, Z) \, \pi(x_{1:t-1}^{(i)} \mid Z)} \\
&= \frac{p(x_t \mid x_{1:t-1}^{(i)}, Z)}{\pi(x_t \mid x_{1:t-1}^{(i)}, Z)} \, w(x_{1:t-1}^{(i)}) \\
&= \frac{p(Z \mid x_{1:t-1}^{(i)}, x_t) p(x_t \mid x_{1:t-1}^{(i)})}{p(Z \mid x_{1:t-1}^{(i)}) \pi(x_t \mid x_{1:t-1}^{(i)}, Z)} \, w(x_{1:t-1}^{(i)}) \tag{5}
\end{aligned}
$$

To obtain the last equation, we apply Bayes rule to decompose $p(x_t \mid x_{1:t-1}^{(i)}, Z)$ that interchanges x_t and Z.

As it is often the case in PF applications, we assume that $\pi(x_t \mid x_{1:t-1}^{(i)}, Z) = p(x_t \mid x_{1:t-1}^{(i)})$. Using this simple exploration based proposal the weight recursion in (5) becomes:

$$
\begin{aligned}
w(x_{1:t}^{(i)}) &= w(x_{1:t-1}^{(i)}) \frac{p(Z \mid x_{1:t-1}^{(i)}, x_t) \, \cancel{p(x_t \mid x_{1:t-1}^{(i)})}}{p(Z \mid x_{1:t-1}^{(i)}) \, \cancel{p(x_t \mid x_{1:t-1}^{(i)})}} \\
&= w(x_{1:t-1}^{(i)}) \frac{p(Z \mid x_{1:t-1}^{(i)}, x_t)}{p(Z \mid x_{1:t-1}^{(i)})} \tag{6}
\end{aligned}
$$

By recursive substitution of weights in (6), i.e., by applying (6) to $w(x_{1:t-1}^{(i)}), w(x_{1:t-2}^{(i)}), \ldots, w(x_{1:2}^{(i)})$, we obtain

$$w(x_{1:t}^{(i)}) = w(x_{1:t-2}^{(i)}) \frac{\cancel{p(Z|x_{1:t-1}^{(i)})}}{p(Z|x_{1:t-2}^{(i)})} \frac{p(Z|x_{1:t-1}^{(i)}, x_t)}{\cancel{p(Z|x_{1:t-1}^{(i)})}}$$

$$= w(x_1^{(i)}) \frac{p(Z|x_{1:t-1}^{(i)}, x_t)}{p(Z|x_1^{(i)})} \tag{7}$$

Finally, under the assumption that all particles have the same initial weight $w(x_1^{(i)})$ and the same initial observation probability $p(Z|x_1^{(i)})$ for $i = 1, \ldots, N$, we obtain

$$w(x_{1:t}^{(i)}) = p(Z|x_{1:t-1}^{(i)}, x_t) \tag{8}$$

The weight in (8) represents particle evaluation with respect to shape and other appearance features of the model described in the observation set Z. The intuitive explanation is that a new correspondence x_t added to the sequence of correspondences $x_{1:t-1}^{(i)}$ should increase the similarity of the selected edge fragments in the image to the model object. Thus, the new weight is more informative if evaluated using the extended set of correspondences $x_{1:t}^{(i)}$, and the old weight $w(x_{1:t-1}^{(i)})$ is not needed for the evaluation. For comparison, the corresponding weight update in the standard PF framework ([29]) is

$$w(x_{1:t}^{(i)}) = w(x_{1:t-1}^{(i)}) \, p(z_t|x_{1:t-1}^{(i)}, x_t), \tag{9}$$

where z_t denotes the new observations obtained at time t. Because our observations Z do not have any natural order, Z cannot be expressed as a sequence of observations. We do not make any Markov assumption in the proposed formula (8), i.e., the new state x_t is dependent on all previous states $x_{1:t-1}^{(i)}$ for each particle (i).

Since the proposed PF framework performs sequential filtering, there are two important issues that need to be addressed: setting the initial correspondences $x_1^{(i)}$ for each particle $i = 1, \ldots, N$ (Section 3.1) and the number of particles, which will be determined experimentally. We only mention here that the proposed approach is in some sense robust to the initial correspondences, since it does not matter with which correspondence we start as long as we start at some element of the optimal set of correspondences $\widehat{x}_{1:m} = (\widehat{x}_1, \ldots, \widehat{x}_m)$. Practically, we start at most promising correspondences, which are determined by sampling form a distribution determined by shape similarity between model contour segments and image edge fragments.

The optimization of Eq. 2 is computed in a framework of Sequential Importance Resampling (SIR). We outline now our PF algorithm, which in each iteration, i.e., at every time step t, and for each particle $i = 1, \ldots, N$ executes the 3 steps:

1) **Importance sampling / proposal:** Sample $x_t^{(i)} \sim \pi(x_t|x_{1:t-1}^{(i)}, Z)$ and set $x_{1:t}^{(i)} = (x_{1:t-1}^{(i)}, x_t^{(i)})$.

2) **Importance weighting/evaluation:** An individual importance weight is assigned to each particle according to Eq. 8.

3) **Resampling:** At the sampling stage we sample a lot more followers than the number of particles which is referred to as prior boosting [12, 4] so as to capture multimodal likelihood regions. Thus we have a larger set of particles $\{\mathbf{x}_{1:t}^{(i)}\}_{i=1}^{M}$ where $M > N$ from which we sub-sample N particles and assign the qual weights to all of them as in the standard SIR approach. While SIR requires a justification that it still computes (4) due to the presence of the old weights in (9), which are reset to be be equal to $1/N$ after resampling in SIR, this fact is obvious in the proposed approach, since the old weights are not present in (8).

In order to be able to execute the proposed algorithm, we need to define the proposal distribution $\pi(x_t|x_{1:t-1}^{(i)}, Z) = p(x_t|x_{1:t-1}^{(i)})$ and the posterior distribution $p(Z|x_{1:t}^{(i)})$. We describe their constructions below.

3.1. Proposal Distribution

We use shape similarity to define the initial proposal distribution. In order to achieve scale invariance, each model segments s_k and edge fragments e_j in the image is sampled with the same number of points, e.g., 20 points. We use a novel shape descriptor described in Section 5 to define shape similarity $\psi : S \times E \to \mathbb{R}_+$. By normalizing ψ we obtain the initial proposal distribution $\pi(x_1|Z)$. An example is shown in Fig. 3(right). By sampling (with repetition) from this distribution, we obtain the initial set of particles $x_1^{(i)} \sim \pi(x_1|Z)$ for $i = 1, \ldots, N$.
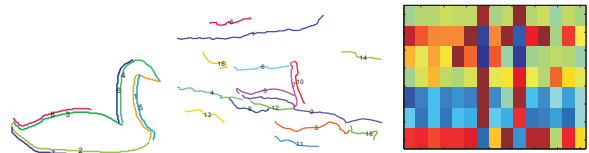

Figure 3. Matrix of similarities between 8 model contour segments left and 16 edge fragments middle. It represents the initial proposal distribution $\pi(x_1|Z)$.

We now describe the proposal distribution for the consecutive integrations of our PF, i.e., $\pi(x_t|x_{1:t-1}^{(i)}, Z) = p(x_t|x_{1:t-1}^{(i)})$. Many object detection methods reported in the literature, e.g., [25], utilize the object centroid as the localization constraint of various object parts. They explore the fact that object parts hinge around the centroid, which significantly reduces object search space when the set of centroid hypotheses is small. Our proposal distribution is based on this idea. We use the shape similarity ψ to define the center point function $CP : S \times E \to I$. It

transforms the center point of the model shape to the image I for a given correspondence $x_1^{(i)} = (s_k, e_j)$, for some k and j and some particle (i). We observe that our model shape has a unique center point. Every possible correspondence transfers it to a object center hypothesis in the image. The centroid transfer is possible, since we estimate the scaling factor from the length ratio of fragments s_k, e_j. Thus, each pair (s_k, e_j) defines a potential center point of the model shape M on the image I. Since each particle $x_{1:t-1}^{(i)} = ((s_1, e_1), \ldots (s_{t-1}, e_{t-1}))$ is a sequence of such pairs, we can extend the definition of the center point to include all particles. Thus, $CP(x_{1:t-1}^{(i)})$ denotes the average center point of the model shape M on the image I. Then, the proposal distribution $p(x_t|x_{1:t-1}^{(i)})$ is defined as a discrete distribution over the set $S \times E$, where the probability of each $x_t \in S \times E$ is proportional to a Gaussian of the distance between $CP(x_{1:t-1}^{(i)})$ and $CP(x_t)$.

Since this proposal distribution is not particularly discriminative in that it is unable to uniquely determine the right extension of a given sequence of corresponding segments $((s_1, e_1), \ldots (s_{t-1}, e_{t-1}))$, due to inaccuracy of the centroid estimation, we determine K followers for each particle $x_{1:t-1}^{(i)}$ by sampling (with repetition) from $p(x_t|x_{1:t-1}^{(i)})$. Thus, sampling form the proposal distribution increases the number of particles to $M = KN$ from N. We recall that in the resampling step, the number of particles is again reduced to N. In all our experimental results $K = 25$ and $N = 150$.

3.2. Likelihood

We define in this section the likelihood $p(Z|x_{1:t}^{(i)})$, which is needed for particle evaluation. We recall that $x_{1:t}^{(i)} = ((s_1, e_1), \ldots (s_t, e_t))$, i.e., it is a sequence of pairs of corresponding model segments and edge fragments. It is defined by similarity between the shape formed by segments of model contour M and the shape formed by the edge fragments

$$p(Z|x_{1:t}^{(i)}) \propto \psi(\bigcup_{j=1}^{t} s_j, \bigcup_{j=1}^{t} e_j), \qquad (10)$$

where ψ is defined in Section 5.

For small t ($t = 1, 2$) this posterior is not particularly discriminative. However, already starting with $t = 3$ or 4, the shape of correctly selected edge fragments starts to resemble the model contour, and consequently, the descriptive power of $p(Z|x_{1:t}^{(i)})$ increases significantly.

4. Contour Models with Part Bundles

Given a single model contour that can be hand drawn or extracted from an example image, we first decompose it

into possibly overlapping model contour parts (or segments) $S = \{s_1, \ldots, s_m\}$; breaking segments at high curvature points. The segments are then grouped into part bundles.

An example bundle decomposition is shown in Fig. 4. In addition to longer contour segments, we need to select shorter ones, since contour parts may be missing in edge images. The main constraint for the bundle design is to ensure that a rough shape sketch obtained by selecting one part form each bundle still resembles the model contour. A bundle can have fragments representing overlapping parts thus allowing for redundancy. A cognitive motivation behind our bundle decomposition scheme is that an object can be recognized even if some parts of it are missing, as can be observed in Fig. 1. There are several reasons why parts of objects can be missing in real images: missing edge information, occlusion, failures in contour grouping. The selection of parts and their grouping into bundles was designed manually. We have one model per shape class and select model parts and grouping them into bundles. However, when ground truth images with detected contour fragments were available, automatic learning part bundles is also possible.
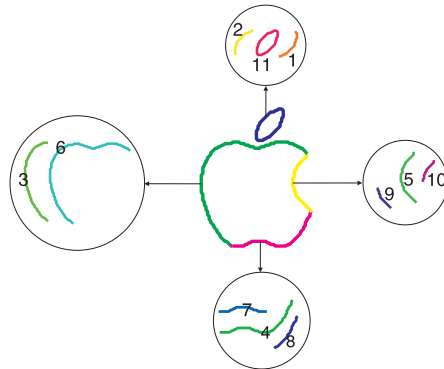

Figure 4. The contour model of the apple and the corresponding part bundles. The contour is shown in the center. The 11 contour fragments are decomposed into four part-bundles.

Formally, $\mathcal{B} = \{B_k\}_{k=1}^{m'}$, where $B_k \subset S$ and $m' \leq m$, is a part bundle decomposition of S if and only if $\bigcup \mathcal{B} = S$ and $B_i \bigcap B_j = \emptyset$ for $i, j = 1, \ldots, m'$ and $i \neq j$.

The part bundles are naturally integrated in our PF framework in that they constrain the proposal distribution $p(x_t|x_{1:t-1}^{(i)})$ defined in §3.1. Given a particle $x_{1:t-1}^{(i)} = ((s_1, e_1), \ldots (s_{t-1}, e_{t-1}))$, at steps $t = 2, \ldots, m' - 1$ we constrain the correspondence $x_t = (s_t, e_t)$ to select s_t that belongs to a different part bundle from the part bundles of $s_1, \ldots, s_{t-1}$. Thus, we ensure to first have one segment from each part bundle. Only when this is satisfied for $t = m'$, we allow to select multiple segments form the same bundles. Intuitively this means that we enforce our particles to first trace a rough shape sketch of the model shape in the edge image before filling in shape details. We show

an example evolution of particle filter in Fig. 5. Matching edge fragments are numbered with corresponding model segments shown in Fig. 4. Rough sketch (matching model segments are from different part bundles) is obtained after iteration 3, shape details are added in iterations 4 and 5.

(a) an image   (b) edge fragments   (c) pf initialization

(d) 1st iteration   (e) 2nd iteration   (f) 3rd iteration   (g) 4th iteration   (h) 5th iteration
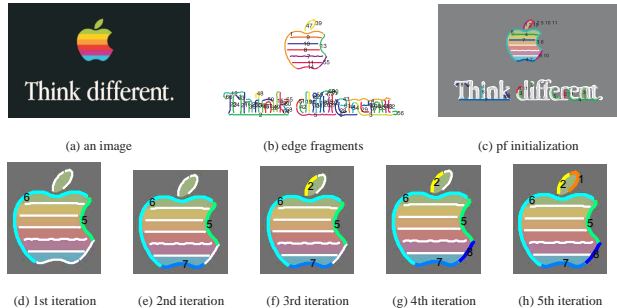
Figure 5. The evolution of particles: (b) shows the edge fragments of (a); (c) edge fragments that are parts of initial particles are in color; (d) to (g) shows edge fragments of particles with highest weights after each iteration. The part bundle model is shown in Fig. 4. Rough sketch is obtained after iteration 3, shape details are added in iterations 4 and 5.

5. Shape Descriptor

We introduce a very simple and intuitive shape descriptor. It can be computed for any set of points X on the plane. (In this paper we apply it to compare sets of contour fragments.) Given a point $A \in X$, a shape descriptor of point A denoted $S_X(A)$ is a histogram of all triangles spanned by A and all pairs of points $B, C \in X$, where points A, B, C must be different. To be more specific, with reference to Fig. 6, $S_X(A)$ is a 3D histogram of the angles BAC, and two distances AB and AC. In order to distinguish the two distances, we require that the triangle BCA is oriented clockwise. In order to make our descriptor scale invariant, the distances are normalized by the average pairwise distance of points in X. The shape descriptor $S(X)$ of the set X is a joint 3D histogram of all points, i.e., $S(X) = \sum \{S_X(x) | \ x \in X\}$. Then, the similarity $\psi(X, Y)$ between two sets X, Y is obtained by the standard histogram intersection.

Our shape descriptor has been inspired by Carlsson [3] (see also [30]), but it is different. Carlson considers only qualitative orientation of each triangle: oriented clock or counterclockwise. Our descriptor provides a full quantitative description of each triangle. This leads to a significant increase in descriptive power. The comparison of our triangle histogram shape similarity measure to other measures is left out due to limited space. We only demonstrate in §6 that it is more flexible than shape context [1], which has been used to evaluate similarity between contour segments in object detection [34]. Shape context considers pairwise relationship between points, while we consider relations between triples of points.
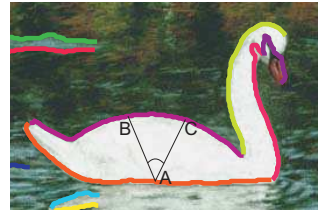

Figure 6. A triangle constructed by points A, B and C.

6. Experimental Results

We present results on the ETHZ shape classes ([10]). It has 5 different object categories with 255 images in total. All categories have significant intra-class variations, scale changes, and illumination changes. Moreover, many objects are surrounded by extensive background clutter and have interior contours. This dataset comes with ground truth gray level edge maps, which is a very important factor for fair comparison, in particular for contour based methods.

Fig. 7 shows P/R curves for three methods: Contour Selection by Zhu et al. [34], Ferrari et al. [8], and our method. We selected these two methods for comparison, since they also are contour based methods, and direct comparison is possible, since [34] published P/R curves in the paper and [8] published their code. We first choose the same criterion that is used in [8] and [34], i.e., a detection is deemed as correct if the detected bounding box covers over 20% of the ground truth bounding box. Our approach performs better than [8] on four categories (exception: "Mugs") and also outperforms [34] on four categories (exception: "Bottles").
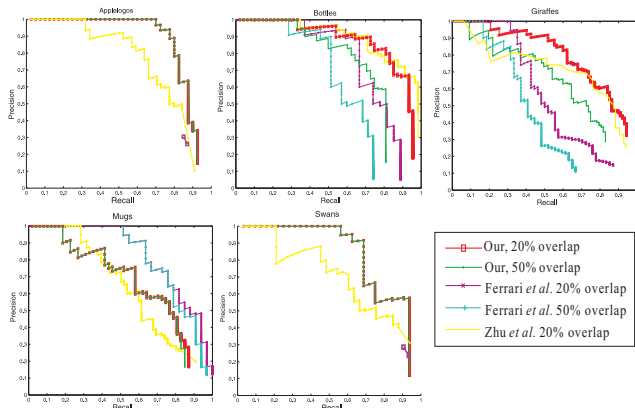

Figure 7. Precision/Recall curves of our method compared to Zhu et al. [34] and the method by Ferrari et al. [8] on ETHZ shape classes. We report both 20% and 50% overlap results whenever available.

Similar to the experimental setup in [34], we use only the single hand-drawn models provided for each class.

Since the criterion of 20% overlap may not indicate a true detection, we also show the results with 50% overlap, which is a standard measure on PASCAL collection. Our P/R curves with 20% and 50% overlap are identical for "Ap-

plelogos" and "Swans". The performance of our system did not change much with 50% criterion for "Mugs". For "Bottles" and "Giraffes" we notice a drop with 50% overlap, but still our performance is better than that of [8].

The 50% overlap results are not reported in [34] and in [8]. However, by running the released code of [8], we are able to report P/R results with both 20% and 50% overlap on the classes "Bottles", "Giraffes" and "Mugs". In [8] only detection rate (DR) vs. false positive per image (FPPI) is reported. Since we were not able to successfully run the code on "Swans" and "Applelogos, for these two classes, we report the translation of their results into P/R from [34].

From the P/R curves we found that our method performs significantly better than the other two methods on non-rigid objects: "Swans" and "Giraffes". We benefit here from our novel shape descriptor. Thin-Plate Spline Robust Point Matching algorithm (TPS-RPM) is used to fine tune the detected contour in [8]. [34] uses shape context as shape descriptor. To illustrate the benefits of our new shape descriptor in the presence of noise and deformation we compare it with shape context (SC) [1] on Kimia99 dataset [24] in Table 1. This dataset has a lot of intra-class deformation.

	1_{st}	2_{nd}	3_{rd}	4_{th}	5_{th}	6_{th}	7_{th}	8_{th}	9_{th}	10_{th}
SC	97	91	88	85	84	77	75	66	56	37
Our	99	97	96	97	96	93	93	88	86	68

Table 1. Retrieval results on Kimia99 dataset

We use 150 particles, $K = 25$ nearest neighbors for proposal distribution. For shape descriptor, we select 6 distance bins (in log space), and 12 angle bins (between 0 and π).

We also use detection rate vs false positive per image (DR/FPPI) in Fig. 8 to evaluate our results. We quote the other curves from [9], which is a longer version of [8], and compare our system to three different methods: [10], [9], and Chamfer matching, also reported in [9]. All the methods use 20% bounding box overlap. From the results we can see that our method outperforms all of them on 0.3 FPPI, and is better in four categories (except "Swans") on 0.4 FPPI. Our precisions at 0.3/0.4 FPPI are Applelogos: 92.5/92.5, Bottles:95.8/95.8, Giraffes: 86.2/92.0, Mugs: 83.3/85.4, Swans: 93.8/93.8.

Some detection examples of our method can be found in Fig. 9. Since we group edge fragments, the detected objects are precisely localized, which is in contrast to appearance based sliding window approaches. We also show some false positives in the bottom row.

7. Conclusions

In addition to the well-known sequential filtering benefit of particle filters that implements delayed decision in a sound statistical framework, one of the main benefits of the proposed PF framework for grouping of edge fragments is the fact that global shape similarity can be explicitly employed. It measures how similar the edge fragments of each
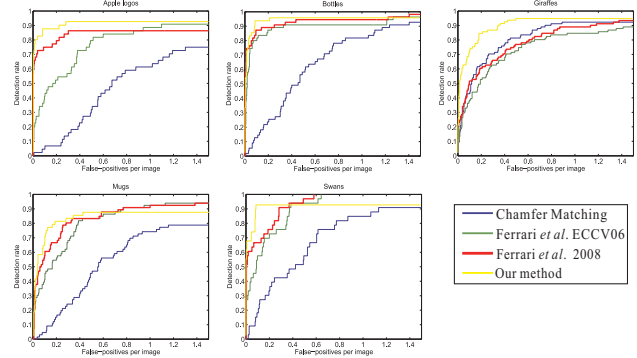

Figure 8. DR/FPPI curves of our method compared to results reported in Ferrari et al [9].

Figure 9. Sample detection results. The edge map is overlaid on the image in white. The detected fragments are shown in black. The corresponding model parts are shown in top-left corners. The red frame in the bottom row shows some false positives.

particle are to the model contour. Thus, providing strong likelihood function for evaluation of each particle. Since each particle carries a contour hypothesis, the proposed approach can handle large variations of object contours including nonrigid deformation and missing parts in cluttered images.

The main limitation of the proposed system is that it works with edge fragments, which are obtained by bottom-up, low level linking of edge pixels, and therefore, it heavily relies on good edge detection results. We assume that the occluding contour of a target object is composed of no more than 10 to 20 edge fragments, which can be broken, deformed, and some parts can be missing. With the recent progress in edge detection, e.g., pb edge detector [20], good

edge detection results on many images are possible, e.g., on the ETHZ dataset [10], and consequently our assumption is satisfied. However, still on many images the performance of edge detectors is unsatisfactory, i.e., our assumption is not satisfied, e.g., the occluding contour of a target object is composed of more than 20 edge fragments that are only few pixels long. This is the main reason why we do not report any experimental results on PASCAL challenge collection of data sets. While on many images in the ETHZ collection, edge detection performs sufficiently well so that our assumption is satisfied, this is not the case for a large percentage of PASCAL images. For a large part this is due to low resolution of these images.

8. Acknowledgments

References

[1] S. Belongie, J. Puzhicha, and J. Malik. Shape matching and object recognition using shape contexts. *IEEE PAMI*, 24:509–522, 2002. 2, 6, 7

[2] G. Borgefors. Hierarchical chamfer matching: A parametric edge matching algorithm. *IEEE PAMI*, 10(6):849–865, 1988. 2

[3] S. Carlsson. Order structure, correspondence and shape based categories. In *In Shape Contour and Grouping in Computer Vision*. 6

[4] J. Carpenter, P. Clifford, and P. Fearnhead. Building robust simulation-based filters for evolving data sets. Technical report, Dept. of Statistics, University of Oxford, 1999. 4

[5] P. Felzenszwalb and D. McAllester. A min-cover approach for finding salient curves. In *CVPRW: Proceedings of the Conference on Computer Vision and Pattern Recognition Workshop*, 2006. 2

[6] P. F. Felzenszwalb and J. Schwartz. Hierarchical matching of deformable shapes. In *CVPR*, 2007. 2

[7] V. Ferrari, L. Fevrier, F. Jurie, and C. Schmid. Groups of adjacent contour segments for object detection. *IEEE PAMI.*, 30(1):36–51, 2008. 2

[8] V. Ferrari, F. Jurie, and C. Schmid. Accurate object detection with deformable shape models learnt from images. In *CVPR*, pages 1–8, 2007. 6, 7

[9] V. Ferrari, F. Jurie, and C. Schmid. From images to shape models for object detection. *INRIA Technical Report*, July 2008. 7

[10] V. Ferrari, T. Tuytelaars, and L. J. V. Gool. Object detection by contour segment networks. In *ECCV*, 2006. 6, 7, 8

[11] M. Galun, R. Basri, and A. Brandt. Multiscale edge detection and fiber enhancement using differences of oriented means. In *ICCV*, 2007. 1

[12] N. Gordon, D. Salmond, and A. Smith. Novel approach to nonlinear/non-gaussian bayesian state estimation. In *Radar and Signal Processing, IEE Proceedings of*, volume 140, pages 107–113, 1993. 4

[13] W. E. L. Grimson. The combinatorics of object recognition in cluttered environments using constrained search. *Artif. Intell.*, 44(1-2):121–165, 1990. 3

[14] W. E. L. Grimson. *Object recognition by computer: the role of geometric constraints*. MIT Press, Cambridge, MA, USA, 1990. 3

[15] W. E. L. Grimson and T. Lozano-Pérez. Localizing overlapping parts by searching the interpretation tree. *IEEE PAMI*, 9(4), 1987. 3

[16] D. Hoiem, A. Stein, A. A. Efros, and M. Hebert. Recovering occlusion boundaries from a single image. In *ICCV*, 2007. 2

[17] P. D. Kovesi. MATLAB and Octave functions for computer vision and image processing, 2008. Available from: <http://www.csse.uwa.edu.au/∼pk/research/matlabfns/>. 3

[18] B. Leibe, E. Seemann, and B. Schiele. Pedestrian detection in crowded scenes. In *CVPR*, 2005. 2

[19] S. Maji and J. Malik. Object detection using a max-margin hough transform, 2009. CVPR. 2

[20] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *ICCV*, 2001. 1, 7

[21] G. McNeill and S. Vijayakumar. Part-based probabilistic point matching using equivalence constraints. In *NIPS*, pages 969–976, 2006. 2

[22] A. Opelt, A. Pinz, and A. Zisserman. A boundary-fragment-model for object detection. In *ECCV*, 2006. 2

[23] S. Ravishankar, A. Jain, and A. Mittal. Multi-stage contour based detection of deformable objects. In *ECCV*, 2008. 2

[24] T. B. Sebastian, P. N. Klein, and B. B. Kimia. Recognition of shapes by editing their shock graphs. *PAMI*, 26(5):550–571, 2004. 7

[25] J. Shotton, A. Blake, and R. Cipolla. Contour-based learning for object detection. In *ICCV*, 2005. 2, 4

[26] A. Stein, D. Hoiem, and M. Hebert. Learning to find object boundaries using motion cues. In *ICCV*, 2007. 2

[27] A. Tamrakar and B. B. Kimia. No grouping left behind: From edges to curve fragments. In *ICCV*, 2007. 2

[28] A. Thayananthan, B. Stenger, P. H. S. Torr, and R. Cipolla. Shape context and chamfer matching in cluttered scenes. In *CVPR*, pages 127–133, 2003. 2

[29] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. The MIT Press Cambridge, 2005. 4

[30] J. Thureson and S. Carlsson. Appearance based qualitative image description for object class recognition. In *ECCV*, 2004. 6

[31] N. Trinh and B. B. Kimia. A symmetry-based generative model for shape. In *ICCV*, 2007. 2

[32] L. Wang, J. Shi, G. Song, and I. Shen. Object detection combining recognition and segmentation. In *ACCV07*, pages 189–199, 2007. 2

[33] Q. Zhu, G. Song, and J. Shi. Untangling cycles for contour grouping. In *ICCV*, 2007. 2

[34] Q. Zhu, L. Wang, Y. Wu, and J. Shi. Contour context selection for object detection: A set-to-set contour matching approach. In *ECCV*, 2008. 2, 6, 7