2D Nonrigid Partial Shape Matching Using MCMC and Contour Subdivision

Yu Cao[1], Zhiqi Zhang[1], Irina Czogiel[2], Ian Dryden[3], Song Wang[1]
[1]Department of Computer Science & Engineering, University of South Carolina, Columbia, SC 29208
[2]Max Planck Institute for Molecular Genetics, Berlin, Germany
[3]Department of Statistics, University of South Carolina, Columbia, SC 29208
{cao, zhangz}@cec.sc.edu, czogiel@molgen.mpg.de, dryden@mailbox.sc.edu, songwang@cec.sc.edu

Abstract

Shape matching has many applications in computer vision, such as shape classification, object recognition, object detection, and localization. In 2D cases, shape instances are 2D closed contours and matching two shape contours can usually be formulated as finding a one-to-one dense point correspondence between them. However, in practice, many shape contours are extracted from real images and may contain partial occlusions. This leads to the challenging partial shape matching problem, where we need to identify and match a subset of segments of the two shape contours. In this paper, we propose a new MCMC (Markov chain Monte Carlo) based algorithm to handle partial shape matching with mildly non-rigid deformations. Specifically, we represent each shape contour by a set of ordered landmark points. The selection of a subset of these landmark points into the shape matching is evaluated and updated by a posterior distribution, which is composed of both a matching likelihood and a prior distribution. This prior distribution favors the inclusion of more and consecutive landmark points into the matching. To better describe the matching likelihood, we develop a contour-subdivision technique to highlight the contour segment with highest matching cost from the selected subsequences of the points. In our experiments, we construct 1,600 test shape instances by introducing partial occlusions to the 40 shapes chosen from different categories in MPEG-7 dataset. We evaluate the performance of the proposed algorithm by comparing with three well-known partial shape matching methods.

1. Introduction

Inherent in object boundaries, structural shape plays an important role in computer vision and object recognition. A fundamental problem is to match different shape instances and measure their shape similarity (or dissimilarity). In the 2D case, each shape instance can be represented by a closed contour that delineates the boundary of the structure of interest. Shape matching can usually be formulated as finding a one-to-one dense point correspondence between two shape contours. In many applications, the resulting correspondence is expected to reflect the possible non-rigid shape deformation between the two shape contours. For example, Figure 1(a) and (b) show two shape contours and their non-rigid shape matching described by a set of corresponded landmark points, where points A, B, C and D match points A', B', C' and D', respectively. The corresponded landmark points may also contain other landmark points on the contours.

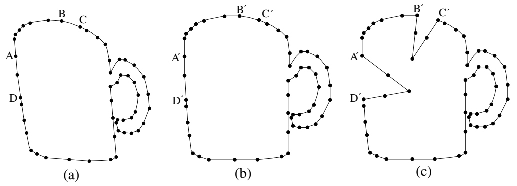

Figure 1. An illustration of the 2D shape matching and partial shape matching. (a) A template shape contour, (b) a target shape contour without partial occlusions, and (c) a target shape contour with partial occlusions.

In practice, however, the object of interest may be occluded by other objects or cropped by image perimeters. In this case, the extracted shape contours may contain partial occlusions. Partial shape matching is a challenging problem that aims to match a partially occluded shape contour to a template shape contour without occlusion by identifying the matched portions from these two contours. An example is illustrated in Fig. 1, where some partial occlusions on the shape contour shown in Fig. 1(b) lead to a shape contour shown in Fig. 1(c). Our goal is to find a match between this occluded shape contour, shown in Fig. 1(c), and a template shape contour shown in Fig. 1(a), by identifying the non-occluded contour segments $A'B'$ and $C'D'$ in Fig. 1(c) that matches the contour segments AB and CD in Fig. 1(a), respectively.

Different from full shape matching [20, 9, 1, 16, 15], the number, the size and the location of the matched contour segments are usually all unknown in partial shape matching. A shape matching cost function that considers all these unknowns is highly non-linear and needs advanced optimization algorithms to find reasonable solutions. For example, some early work on partial shape matching uses simulated annealing and genetic algorithms to search for an optimal solution [5, 17], where each shape contour is represented by a set of line segments and the matching is based on some features extracted from these segments. In [2], local invariant features based on curvature or cumulative turn angle of curves are derived for partial shape matching. Recent work along this direction includes [18, 22]. These methods assume high similarity between the matched segments and may not describe well the global non-rigid deformation between the two shape contours.

Recently, Latecki *et al*. [14, 13] developed an elastic partial shape matching algorithm to model a possible non-rigid shape deformation. In this approach, the problem is formulated as identifying a continuous segment from a target shape contour such that this identified contour segment best matches a query contour segment. Non-rigid shape deformation is considered in the contour-segment matching and a shortest-path-like algorithm is developed to efficiently search for an optimal solution. Note that the problem handled in [14, 13] is different from our problem from two perspectives. First, the query (or template) shape contour is a contour segment that will be fully matched to a segment of the target contour, while in our problem both template and target are full contours and the matched contour segments need to be identified from both template and target shape contours. Second, in [14, 13], only one continuous contour segment is identified for matching the template segment, while in this paper we may get multiple matched segments that are not connected, as shown in Fig. 1(c).

More closely related to this paper is a partial shape matching method recently developed by Chen, Feris and Turk [3]. In this method, two shape contours are first sampled at a sequence of points and various features are extracted from each sampled point. This way, the partial shape matching problem is reduced to the molecular subsequence matching problem studied in computational biology. In [3], the Smith-Waterman algorithm [21], which is widely used for molecular subsequence matching, is used to achieve efficient partial shape matching.

In this paper, we propose a new algorithm for mildly non-rigid partial shape matching by using MCMC (Markov chain Monte Carlo) simulation. Similar to [3], we represent each shape contour by a sequence of landmark points. The selection of subsequences of these landmark points into the shape matching is evaluated and updated by a posterior, which is composed of both a matching likelihood and a prior. This prior favors the inclusion of more and consecutive landmark points into the matching. The matching likelihood is defined by the full Procrustes distance [7] between the selected subsequences of landmark points. To better describe the matching likelihood, we develop a contour-subdivision technique to highlight the contour segment with highest matching cost from the selected subsequences of the landmark points. And the matching cost of the highlighted contour segment will contribute to the speed-up of the convergence of the MCMC algorithm to an optimum matching solution. In our experiments, we construct $1,600$ test shape instances by introducing partial occlusions to the 40 shapes chosen from different categories in the MPEG-7 dataset. We evaluate the performance of the proposed algorithm by comparing with 3 well-known shape matching methods which are 'Elastic Matching'[14, 13], 'Shape Contexts'[1] and 'Smith-Waterman'[3].

The remainder of the paper is organized as follows. Section 2 describes the details of the proposed algorithm. Section 3 describes the experiment design, results and performance evaluation. Section 4 is the conclusion.

2. The Proposed Method

Our model for 2D partial shape matching is built upon a Procrustes-based likelihood with match matrix parameters indicating which parts of the contours match. A prior distribution is specified for the unknown parameters, and then Bayesian inference is carried out by simulating from the posterior distribution using Markov chain Monte Carlo (MCMC) techniques. Using Bayes theorem we have

$$Posterior \propto Likelihood \times Prior$$

and the posterior distribution of the parameters is of primary interest. A Bayesian estimate of the partial shape match is the posterior mode, but this is very difficult to calculate. However, we can find an approximation to the mode by simulating from the posterior distribution (using MCMC) and then choosing the simulated observation which maximizes the posterior.

2.1. The likelihood and the prior distribution

Given two (closed) shape contours s_1 (target) and s_2 (template) which are both represented by a set of ordered landmark points, we write the complex configuration vectors of s_1 and s_2 as

$$Z_1 \in \mathbf{C}^{m_1} \quad Z_2 \in \mathbf{C}^{m_2}$$

where Z_i are complex m_i-vectors each of non-coincident landmark coordinates, $i = 1, 2$. Here m_1 and m_2 are the number of landmark points which represent the shapes contours s_1 and s_2, respectively. Note that m_1 and m_2 need not be the same.

Let us first consider the situation where $m_1 = m_2 = k$, say. The full Procrustes distance [7] between s_1 and s_2 can be calculated using the following formula

$$d_F(Z_1, Z_2) = \left\{ 1 - \frac{Z_1^* C Z_2 Z_2^* C Z_1}{Z_2^* C Z_2 Z_1^* C Z_1} \right\}^{1/2} \qquad (1)$$

where $C = I_k - \frac{1}{k} 1_k 1_k^T$ is the $k \times k$ centering matrix, I_k is the $k \times k$ identity matrix, 1_k is the k-vector of ones, and Z_i^* is the complex conjugate of the transpose of Z_i (a $1 \times k$ complex vector). The full Procrustes distance provides a measure of shape distance between s_1 and s_2, where shape is understood to mean all the geometrical properties that are invariant under translation, rotation and scaling. If s_1 and s_2 are identical up to an arbitrary translation, rotation and scaling then $d_F = 0$. The maximal distance between two shapes is $d_F = 1$.

In the above distance (1), it is assumed that landmark j on shape s_1 corresponds to landmark point j on s_2, for $j = 1, \ldots, k$. However, in the partial shape matching problem we wish to obtain the best correspondence between the landmark points of two shape contours s_1 and s_2, including specifying the matching points and parts of either contour that do not match.

Given that we have closed shape contours, all landmark points are considered ordered (counterclockwise say) with cyclic wrapping. So, for example, the final landmark point is a neighbor of the first landmark point. In order to specify which parts of an object are matching we use a match matrix M [11], which is of dimension $m_1 \times (m_2 + 1)$ and contains either 0 or 1 in each entry. If the (i, j)th element of M is 1 this represents a match from landmark point i on the target to landmark point j on the template. We enforce that there is a single 1 in each row, with 0s representing non-matches. The last column of match matrix M is called the *Dummy Column*. For any target landmark point i, if $M(i, m_2 + 1) = 1$ then the target landmark point i is not included in the matching. In addition, as one progresses down from one row to another, if a '1' is present it must be to the right of the previous '1' in the previous row (up to cyclic wrapping at the ends). This constraint ensures there are no crossings in the correspondence from the target to the template. The number of free parameters in M is $m_1 + m_2 + 1$, since for each shape we have an indicator variable stating if the landmark point is present in the matching or not, and in addition we need to specify where a start landmark point on the target maps to the template.

As there is a single 1 in each row of M, we have the same number of matched landmark points on each shape, which we denote as p. Given a particular match matrix M we can construct the matched landmark points on the target as the p-vector $\hat{Z}_1$ and the corresponding matched landmark points on the template as $\hat{Z}_2$. With the same number of landmark points we can then calculate the Procrustes distance (1) between $\hat{Z}_1$ and $\hat{Z}_2$ (both complex p-vectors), which are the corresponding landmark points indicated by the match matrix.

We introduce the definition of the discrepancy between the matching landmark points as

$$D_{s_1,s_2}(Z_1, Z_2|M) = d_F(\hat{Z}_1, \hat{Z}_2) \qquad (2)$$

where d_F is the full Procrustes Distance function (1). The discrepancy describes the dissimilarity between the subsets from template and target, respectively.

Using the discrepancy measure (2) we define the likelihood of the target given the template and the parameters (M, τ) as

$$L(Z_2|M, \tau, Z_1, \xi) \propto \tau^{\xi - 1} \exp(-\tau D_{s_1,s_2}(Z_1, Z_2|M)) \qquad (3)$$

where $\tau \in \mathbf{R}^+$ is a precision parameter, and $\xi > 1$ is a fixed parameter. This likelihood is similar in spirit to that of [6] and [19] except we have a fixed power for the precision parameter and the discrepancy measure is not squared. Our interpretation is that the model is analogous to an approximate Laplace distribution in the shape tangent space for the matching landmark points and the unmatched landmark points are independently and uniformly distributed on a region with area A/τ, where A is fixed.

Given two shapes s_1 and s_2 represented by a set of ordered landmark points and proposed values for matching parameters, we can calculate the likelihood of the proposed match. We shall also introduce a prior distribution for the parameters. The inclusion of the prior distribution for M allows us to specify our prior beliefs as to how well the model parameters conform to certain desirable properties of the selected and unselected landmark points on s_1 and s_2. The first desirable property is that the larger number of landmark points selected for matching, the better. Without this property, the procedure would favor choosing less and less landmark points since the fewer the number of landmark points, the smaller the full Procrustes distance (1) will tend to be. The second desirable property is that we prefer unselected landmark points to occur in sequence rather than being spread all over the shape contour. Without this property, the algorithm may favor many small matching segments. Therefore we consider the following gap penalty prior density [19]

$$\pi(M|a, b) \propto \exp(-a \cdot GN - b \cdot GL) \qquad (4)$$

here, $a > 0$ and $b > 0$ are the penalty parameters, GN is the number of gaps, and GL is the total gap length. A 'gap' is defined as either a single unmatched landmark point or a set of adjacent unmatched landmark points. In this paper, we sample sparser landmark points on the target than on the template, which usually requires $m_1 < m_2$. Specifically, we intentionally choose $m_2 = 2 \cdot m_1$ for all experiments.

This way, gaps on the target are the desirable ones for measuring GN and GL in Eq. (4).

Furthermore the precision parameter τ has an independent Gamma(α,β) prior.

2.2. Posterior and MCMC Inference

The joint posterior density of the parameters conditional on the given data has the form

$$\pi(M, \tau | Z_1, Z_2) \propto \tau^{\xi+\alpha-1} \cdot$$
$$\exp\{-\tau(D_{s_1,s_2}(Z_1, Z_2|M) + \beta)\} \cdot \exp(-a\; GN - b\; GL)$$

Bayesian inference is then carried out using the posterior distribution. However, the posterior density is very complicated and so we consider Markov chain Monte Carlo (MCMC) simulation to provide approximate values from the posterior distribution [10]. In particular, we use the Metropolis-Hastings algorithm to update the parameters M, and a Gibbs update for τ. The resulting Markov chain of simulated values has the joint posterior as its equilibrium distribution. For the Metropolis-Hastings algorithm at each iteration we have the current value of the parameters M and we propose a new value M^* from a distribution with probability $q(M^*|M)$. The proposed value is accepted with a certain probability α as given in Algorithm 1 where $\pi(M)$ is the probability function of the posterior given the data and τ. For the Gibbs step we simulate from the conditional distribution of τ given all other parameters.

Algorithm 1 is the Metropolis-Hastings algorithm [10] for updating a parameter vector at iteration t, denoted by $\theta^{(t)}$, where $\theta = (M, \tau)$. Note $\pi(\theta)$ is the posterior density for θ given the data Z_1, Z_2 and $q(\theta^{(t)}|\theta^*)$ is the proposal density from moving from θ^* to $\theta^{(t)}$

Algorithm 1 : Metropolis-Hastings algorithm

```
At each iteration $t$.
Sample $\theta^* \sim q(\theta^*|\theta^{(t)})$. Accept with probability
    $\alpha(\theta^{(t)}, \theta^*) = \min\{1, \frac{\pi(\theta^*)q(\theta^{(t)}|\theta^*)}{\pi(\theta^{(t)})q(\theta^*|\theta^{(t)})}\}.$
if acceptance then
    $\theta^{(t+1)} = \theta^*$,
else
    $\theta^{(t+1)} = \theta^{(t)}$.
end if
```

The effectiveness of an MCMC scheme is highly dependent on the choice of proposal distribution. Given a particular set of parameters M we will make a proposed move with probability $q_c(M^*|M)$, where c is the potential number of unmatched landmark points on the template that landmark point i could be matched to, respecting the cyclic order of the matched landmark points on each figure. This is the same as the number of 0s in the current row of M between the previous and next columns which include a 1 (up to cyclic wrapping and ignoring the dummy column).

1. With probability $1/2$ a *match-unmatch* proposal is made. A target landmark point is selected at random. If the landmark point is currently matched it becomes unmatched. If the landmark point is currently unmatched it becomes matched to one of the c potential template candidates with probability $1/c$ (or remains unmatched if $c = 0$).

2. With probability $1/2$ a *match-match* proposal is made. A matched target landmark point is selected at random. The landmark point becomes matched to one of the c potential template candidates with probability $1/c$ (or remains unchanged if $c = 0$).

Some examples of possible proposals are given in Fig. 2. In Fig. 2(a) the ith landmark point is currently matched and $c = 0$ and in Fig. 2(b) it is currently unmatched and $c = 0$. In Fig. 2(c) the ith landmark point is currently matched and $c = 2$ and in Fig. 2(d) it is currently unmatched and $c = 3$.

Note that in our examples the ratios of the posteriors in the Hastings ratio are usually much larger (or much smaller) than the ratio of proposal probabilities, and the MCMC algorithm tends to move towards parameters with high posterior density. For the Gibbs step we simulate the new τ from the conditional distribution

$$\Gamma(\alpha + \xi, \beta + D_{s_1,s_2}(Z_1, Z_2|M)),$$

which is also a Gamma distribution, since the prior for τ is conjugate prior here given the other parameters and the data.

As the Markov chain is irreducible and aperiodic after a large number of iterations the simulated values from the MCMC algorithm are dependent realizations from the approximate posterior distribution. Estimates of interest like the posterior mode, posterior mean, posterior credibility intervals are then calculated from the chain, after ignoring an initial burn-in period.

2.3. Contour Subdivision

By using the full Procrustes distance (1) for defining the likelihood of the shape matching, the above developed MCMC-based partial shape matching algorithm considers the global similarity between the template and the target. However, it does not consider the local shape deformation. As a result, this algorithm may not accurately capture the desirable cut-off points which separate the matched and unmatched contour segments. The major reason is that the inclusion of a small number of undesirable landmark points to the matched segments, or the exclusion of a small number

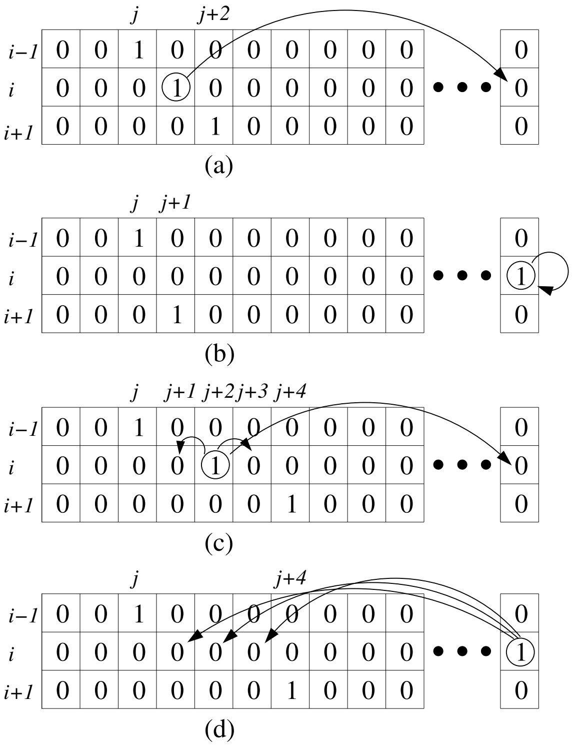

Figure 2. An illustration of possible proposals for MCMC. The last column in each case is the dummy column. (a) $c = 0$ and a proposed change is to make the target landmark point unmatched. (b) $c = 0$ and the only proposal is to make the target landmark point i remain unmatched. (c) $c = 2$ and the proposal possibilities are to make the target landmark point i either match to template landmark point $j + 1$ or $j + 3$, or become unmatched. (d) $c = 3$ and the proposal is to make target landmark point i match one of the three template landmark points between j and $j + 4$.

of desirable landmark points out of the matched segments, do not change the value of the full Procrustes distance (1) much, which is dominated by the majority of the matched landmark points.

We propose to address this problem by recursively dividing the matched contour segments into smaller subsequences and then measure the discrepancy between the matching contour segments not only globally, i.e., the full Procrustes distance between all matched landmarks, but also locally, i.e., the full Procrustes distances between each possible matched segment subsets. By applying this contour division technique, it will also contribute to the speed-up of the convergence of MCMC inference. This strategy is inspired by the hierarchical shape decomposition approaches developed in [8] and [16], where the goal is to find an overall shape matching measure for shape retrieval.

Specifically, starting from the (ordered) matching landmark points $\hat{Z}_1$ and $\hat{Z}_2$, we calculate their full Procrustes distance. We then sequentially and uniformly divide $\hat{Z}_i$ into two ordered landmark sets $\hat{Z}_i^{(1)}$ and $\hat{Z}_i^{(2)}$, $i = 1, 2$, and calculate the Procrustes distances between the subsets $\hat{Z}_1^{(j)}$ and $\hat{Z}_2^{(j)}$, $j = 1, 2$. This is performed recursively and the Procrustes distances between further subdivided landmark point subsets are also calculated. The largest one among all the calculated Procrustes distances are then chosen as the new definition of the discrepancy between $\hat{Z}_1$ and $\hat{Z}_2$ and used for calculating the matching likelihood in Eq. (3). The recursion is finished when the length of the subdivided contour segments reaches a preset threshold. In practice, we also consider different ways of dividing $\hat{Z}_1$ and $\hat{Z}_2$ with cyclic wrapping. Assuming all landmark points are uniformly sampled on both the template and the target, this recursive algorithm can be summarized as follows:

Algorithm 2 $D(s_1, s_2)$ = RecursiveDistPro($\hat{Z}_1$, $\hat{Z}_2$, T)

```
$dist = d_F(\hat{Z}_1, \hat{Z}_2)$.
for $t = 1$ to $T$ do
    $\hat{Z}_1 \leftarrow$ circularly shift $\hat{Z}_1$ by 1.
    $\hat{Z}_2 \leftarrow$ circularly shift $\hat{Z}_2$ by 1.
    $temp$ = RecursiveDist($\hat{Z}_1$, $\hat{Z}_2$, $T$).
    if $temp > dist$ then
        $dist = temp$.
    end if
end for
Return $dist$.
```

Algorithm 3 : D = RecursiveDist($\hat{Z}_1$, $\hat{Z}_2$, T)

```
$length \leftarrow$ number of landmarks in $\hat{Z}_1$.
$dist = d_F(\hat{Z}_1, \hat{Z}_2)$.
if $length > T$ then
    $\hat{Z}_1^{(1)} \leftarrow$ first half of the $\hat{Z}_1$.
    $\hat{Z}_2^{(1)} \leftarrow$ first half of the $\hat{Z}_2$.
    $\hat{Z}_1^{(2)} \leftarrow$ second half of the $\hat{Z}_1$.
    $\hat{Z}_2^{(2)} \leftarrow$ second half of the $\hat{Z}_2$.
    $dist_1$ = RecursiveDist($\hat{Z}_1^{(1)}$, $\hat{Z}_2^{(1)}$, $T$).
    $dist_2$ = RecursiveDist($\hat{Z}_1^{(2)}$, $\hat{Z}_2^{(2)}$, $T$).
    $dist = \max\{dist, dist_1, dist_2\}$.
end if
Return $dist$.
```

Note that a Bayesian inference with a similar matching matrix M is used in [19] for matching the molecular structures of proteins. However, the method developed in this paper is different from the method in [19] from many perspectives: 1) We use recursive Procrustes distances for measuring matching distance, 2) the proposed method deals with mild non-rigid deformations, 3) we target at applications different from [19] – we match closed shape contours while [19] does not, and 4) the MCMC algorithm settings in the proposed method are different from those used in [19].

3. Experiments

In many previous papers [22, 14], the performance of a partial shape matching algorithm is usually evaluated by using it for shape recognition and shape retrieval. In such cases, the performance of a shape matching algorithm is better if it leads to larger shape-matching cost between shape contours drawn from different shape categories and smaller shape-matching cost between the shape contours drawn from the same shape category. This indirect evaluation strategy is highly dependent on the test data: inaccurate shape matching results may be sufficient for shape classification if the shapes to be classified are of small number and are substantially different from each other. In addition, shape classification and retrieval are not the only applications of shape matching. For example, accurate shape matching is an important step in constructing a statistical shape model of a structure [4]. In this section, we develop an approach that is independent of shape classification and retrieval and more directly evaluates the performance of partial shape matching.

In our experiments, we use a set of real shape contours as the template and for each template, we construct a large set of non-rigidly deformed and partially-occluded shape contours in a synthetic way. This way, the ground-truth for the shape matching is known, which can be used to conduct a more objective performance evaluation. Specifically, we pick one shape contour from each of the 40 shape categories in the MPEG-7 dataset [12] as the template. For each template shape contour, we construct 40 target shape contours by adding random (mild) deformations and partial occlusions. To generate a shape contour, the random deformation is added by repeating the following two sequential steps three times: (a)rotating the contour by an angle drawn from a uniform distribution in the range of $[0, 2\pi)$, and (b)scaling the contour along x- and y-directions, where the scaling factors along these two directions are *independently* drawn from a uniform distribution in the range of $[0.8, 1.2]$. Starting from the deformed contour $\tilde{S}$, partial occlusions are added by, first, removing randomly selected n segments from $\tilde{S}$, with n being a random integer uniformly drawn from $1, 2, 3$ and 4. Second, filling the n gaps along $\tilde{S}$ by other contour segments randomly cropped from other shape contours in MPEG-7 that are of a different shape category from the template of $\tilde{S}$. The resulting contour S will be included into our test data set as a target shape contour if the total length of the newly filled segments in S counts for no more than 40% of the total length of S.

In total, we construct $40 \times 40 = 1,600$ target shape contours. Figure 3 shows some data samples of our constructed dataset.

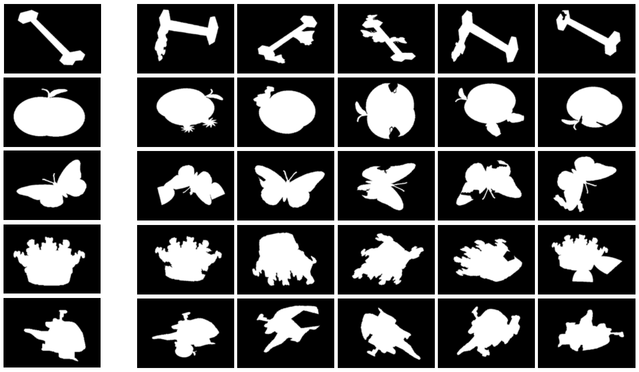

Figure 3. An illustration of the data samples of the constructed dataset. The first column shows 5 template samples from different class categories in the constructed dataset. From column 2 to 6, each row shows 5 instance samples with nonrigid deformations and occlusions.

We then match each constructed shape contour against its original template contour using the developed partial shape matching algorithm. As introduced earlier, we need denser landmark points on the template than these on the target. In this experiment, we uniformly sample 256 landmark points on each template and 128 landmarks on each target shape contour. The total iterations of the MCMC inference is set to be $70,000$. In the first $50,000$ iterations, the likelihood is calculated based on original full Procrustes distance. The penalty parameters for the gap length and the gap number in Eq. (4) are set to $a = b = 3$. The initial value of τ is set to be 50. And the hyper-parameters for the prior of τ are set to be $\alpha = 90$ and $\beta = 0.035$, as we are expecting a small shape distance. In the remaining $20,000$ iterations, the likelihood is calculated based on our contour subdivision technique. In these $20,000$ iterations, the only difference in the parameters is that we set $\alpha = 50$ and $T = 8$. Our contour subdivision strongly relies on a reasonably good temporary matching which is provided by the first $50,000$ iterations. Without the first $50,000$ iterations, the contour subdivision will make the MCMC algorithm converge very slowly. In addition, we apply a technique similar to the contour subdivision to the matching results. It will further cut off the matched segments on the template contour and target contour with a dissimilarity larger than certain threshold. The success of this cut-off technique highly depends on the quality of MCMC matching results.

Figure 4 shows eight examples of partial shape matching resulting from the proposed algorithm. The target (in blue) is shown inside the template contour (in green) and the matched landmark point pairs are linked by red lines. Here we intentionally scale the template and put it inside the target to help illustrate the matching in this figure. In fact, it would be no problem even if we zoom out or zoom in any one of them since the proposed algorithm is invariant to scaling deformation.

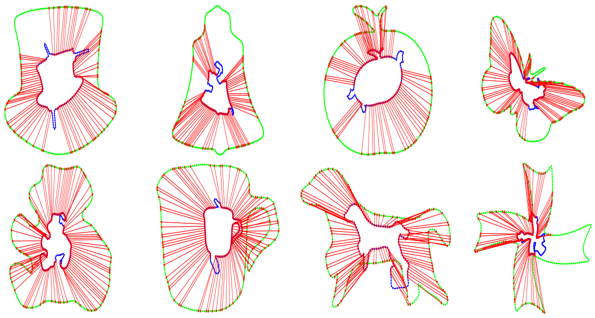

Figure 4. Sample results from the proposed partial shape matching algorithm. The target (in blue) is shown inside the template (in green). The matched landmark point pairs are linked by red lines.

We can quantitatively measure a shape matching score by comparing with the known ground truth. Particularly, in constructing a target shape contour S, we know that S_t represents the contour segments inherent from the (deformed) template and $\bar{S}_t$ represents the contour segments newly introduced to simulate occlusions. Let S_d be the segments of S that are identified by a partial shape matching algorithm to have matchings in the template, and $\bar{S}_d$ be the unmatched segments in S according to this algorithm. We can define the shape-matching score as

$$\rho(S) = \frac{|S_t \cap S_d| + |\bar{S}_t \cap \bar{S}_d|}{|S|}, \tag{5}$$

where $|\cdot|$ measures the length of the segments. This score definition combines both the precision and recall and takes values in the range of $[0, 1]$. The larger this score, the better the matching result. In our experiment, we have 40 target shape contours for each template. We calculate the matching score for each of them and then take the average over all 40 of them as the average matching score for this template shape.

In this experiment, we compare the proposed method with three existing shape matching methods: 'Elastic Matching'[14, 13], 'Shape Contexts'[1], and 'Smith-Waterman'[3]. We implement 'Smith-Waterman' by following the details introduced in [3]. And for 'Shape Contexts' and 'Elastic Matching', we use their original code with default parameters. For 'Smith-Waterman' and 'Elastic Matching' methods, 128 landmark points are uniformly sampled from each shape contour. As for 'Shape Contexts', in order to make this full shape matching method applicable to our partial shape matching problem, we sample 128 landmark points on each template contour and 256 landmark points on each target contour. This way, 'Shape Contexts' may match a subset of landmark points on the target contour and this leads to a partial shape matching.

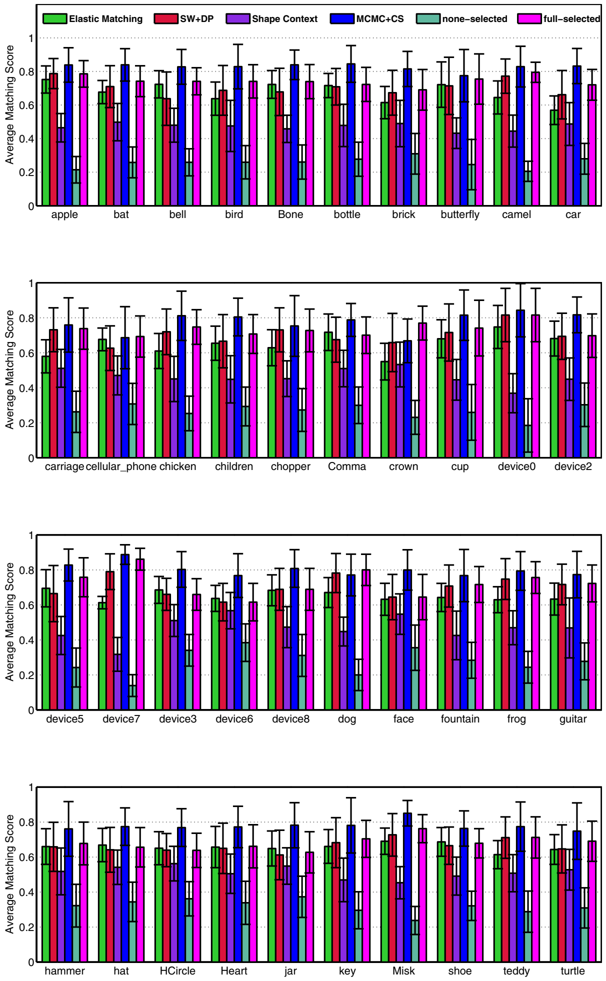

Figure 5. The performances of the 'MCMC+CS'(MCMC + Contour Subdivision), 'SW+DP'(Smith-Waterman + Dynamic Programming), 'Elastic Matching', 'Shape Contexts' and two special cases on 40 shape categories.

Figure 5 shows the average matching scores (and the standard deviation) in terms of each shape category. Clearly, our proposed method (MCMC + Contour Subdivision) outperforms the three comparison methods. Besides the performance of 'MCMC+CS' (MCMC + Contour Subdivision), 'SW+DP'(Smith-Waterman + Dynamic Programming), 'Elastic Matching' and 'Shape Contexts', we also include the performance for the case of (a) *None-Selected*, where none of the segments in the target is selected for matching, and (b) *Full-Selected*, where the whole target shape contour is selected for matching. In case (a), $S_d = \emptyset$ and the matching score of S is reduced to

$$\rho(S) = \frac{|\bar{S}_t|}{|S|}, \tag{6}$$

and in case (b), $S_d = S$ and the matching score of S is reduced to

$$\rho(S) = \frac{|S_t|}{|S|}, \quad (7)$$

Note that, the matching in case (b) to achieve the score shown in Eq. (7) is not trivial: it also assumes the correct matching between the segments S_t to the corresponding segments in the template as in the ground truth. Figure 5 shows that the proposed 'MCMC+CS' method has a performance better than both the *None-Selected* and the *Full-Selected* cases except only in very few categories.

4. Conclusions

In this paper, we developed a new MCMC-based algorithm for partial shape matching. This algorithm considers not only a matching likelihood using global or local Procrustes distance between the selected matching segments, but also a prior that favors longer and more consecutive segments to be included into matching. The proposed approach can match two shape contours where both of them only have subsets of contour segments involved in matching. It can also well handle the non-rigid deformation between the matched shape contours. We also constructed a shape data set with $1,600$ shape contours with deformation and partial occlusions for evaluating the matching performance. We found that the proposed method outperforms the 'Elastic Matching', 'Shape Contexts' and 'Smith-Waterman' algorithms.

Acknowledgments

This work was supported, in part, by NSF IIS-0951754, NSF IIS-1017199, and Army Research Laboratory (ARL) under Cooperative Agreement Number W911NF-10-2-0060. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either express or implied, of NSF, ARL, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes, notwithstanding any copyright notation herein. This research was partially conducted on the computers of the High Performance Computing Group at the University of South Carolina.

References

[1] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *PAMI*, 24(24):509–522, April 2002.

[2] A. Bruckstein, N. Katzir, M. Lindenbaum, and M. Porat. Similarity-invariant signatures for partially occluded planar shapes. *IJCV*, 7(3):271–285, 1992.

[3] L. Chen, R. F. R., and M. Turk. Efficient partial shape matching using Smith-Waterman algorithm. In *CVPR Workshops*, pages 1–6, 2008.

[4] T. Cootes, C. Taylor, D. Cooper, and J. Graham. Active shape models-their training and application. *CVIU*, 61(1):38–59, Jan. 1995.

[5] M. DiIanni, R. Diekrnann, R. Laling, J. Schulze, and S. Tschoeke. Simulated annealing and genetic algorithms for shape detection. *Control and Cybemet.*, 25(1):159–175, 1996.

[6] I. L. Dryden, J. D. Hirst, and J. L. Melville. Statistical analysis of unlabeled point sets: Comparing molecules in chem-informatics. *Biometrics*, 63:237–251, 2007.

[7] I. L. Dryden and K. V. Mardia. *Statistical Shape Analysis*. Wiley, Chichester, 1998.

[8] P. F. Felzenszwalb and J. D. Schwartz. Hierarchical matching of deformable shapes. In *CVPR*, pages 17–22, 2007.

[9] Y. Gdalyahu and D. Weinshall. Flexible syntactic matching of curves and its application to automatic hierarchical classification of silhouettes. *PAMI*, 21(12):1312–1328, December 1999.

[10] W. R. Gilks, S. Richardson, and D. J. Spiegelhalter. *Markov Chain Monte Carlo in Practice*. Chapman and Hall/CRC, Florida, 1996.

[11] P. J. Green and K. V. Mardia. Bayesian alignment using hierarchical models, with applications in protein bioinformatics. *Biometrika*, 93:235–254, 2006.

[12] L. Latecki and R. Lakaemper. Shape similarity measure based on correspondence of visual parts. *PAMI*, 22(10):1–6, October 2000.

[13] L. Latecki, Q. Wang, S. Koknar-Tezel, and V. Megalooikonomou. Optimal subsequence bijection. *ICDM*, pages 565–570, 2007.

[14] L. J. Latecki, V. Megalooikonomou, Q. Wang, and D. Yu. An elastic partial shape matching technique. *Pattern Recognition*, 40(11):3069–3080, 2007.

[15] W.-Y. Lin, N. Boston, and Y. H. Hu. Summation invariant and its applications to shape recognition. In *ICASSP*, volume 5, pages 205–208, 2005.

[16] G. McNeill and S. Vijayakumar. Hierarchical procrustes matching for shape retrieval. In *CVPR*, pages 885–894, 2006.

[17] E. Ozcan and C. K. Mohan. Partial shape matching using genetic algorithms. *Pattern Recognition Letters*, 18:987–992, 1997.

[18] E. Saber, Y. Xu, and A. Tekalp. Partial shape recognition by sub-matrix matching for partial matching guided image labeling. *Pattern Recognition*, 38:1560–1573, 2005.

[19] S. C. Schmidler. Fast Bayesian shape matching using geometric algorithms (with discussion). In *Proc. Valencia/ISBA 8th World Meeting on Bayesian Statistics*, pages 471–490, Benidorm (Alicante, Spain), June 1st-6th 2007.

[20] D. Sharvit, J. Chan, H. Tek, and B. Kimia. Symmetry-based indexing of image databases. *J. VCIR*, 9(4):366–380, December 1998.

[21] T. Smith and M. Waterman. Identification of common molecular subsequences. *Journal of Molecular Biology*, 147:195–197, 1981.

[22] M. Tanase and R. Veltkamp. Part-based shape retrieval. In *ACM Multimedia*, pages 543–546, 2005.