Shape Matching and Recognition – Using Generative Models and Informative Features

Zhuowen Tu and Alan L. Yuille

Departments of Statistics,
University of California, Los Angeles, 90095 USA
{ztu,yuille}@stat.ucla.edu

Abstract. We present an algorithm for shape matching and recognition based on a generative model for how one shape can be generated by the other. This generative model allows for a class of transformations, such as affine and non-rigid transformations, and induces a similarity measure between shapes. The matching process is formulated in the EM algorithm. To have a fast algorithm and avoid local minima, we show how the EM algorithm can be approximated by using *informative features*, which have two key properties–*invariant* and *representative*. They are also similar to the proposal probabilities used in DDMCMC [13]. The formulation allows us to know when and why approximations can be made and justifies the use of bottom-up features, which are used in a wide range of vision problems. This integrates generative models and feature-based approaches within the EM framework and helps clarifying the relationships between different algorithms for this problem such as shape contexts [3] and softassign [5]. We test the algorithm on a variety of data sets including MPEG7 CE-Shape-1, Kimia silhouettes, and real images of street scenes. We demonstrate very effective performance and compare our results with existing algorithms. Finally, we briefly illustrate how our approach can be generalized to a wider range of problems including object detection.

1 Introduction

Shape matching has been a long standing problem in computer vision and it is fundamental for many tasks such as image compression, image segmentation, object recognition, image retrieval, and motion tracking. A great deal of effort has been made to tackle this problem and numerous matching criteria and algorithms have been proposed. For example, some typical criteria include Fourier analysis, moments analysis, scale space analysis, and the Hausdorff distance. For details of these methods see a recent survey paper [14].

The two methods most related to this paper are shape contexts [3] and softassign [5]. Shape contexts method is a feature-based algorithm which has demonstrated its ability to match certain types of shapes in a variety of applications. The softassign approach [5] formulates shape registration/matching as

T. Pajdla and J. Matas (Eds.): ECCV 2004, LNCS 3023, pp. 195–209, 2004.

free energy minimization problem using the mean field approximation. Recent improvements to these methods include the use of dynamic programming to improve shape contexts[12] and the Bethe-Kikuchi free energy approximation [9] which improves on the mean field theory approximation used in the softassign [5].

Our work builds on shape contexts [3] and softassign [5] to design a fast and effective algorithm for shape matching. Our approach is also influnced by ideas from the Data-Driven Markov Chain Monte Carlo (DDMCMC) paradigm [13] which is a general inference framework. It uses data-driven proposals to activate generative models and thereby guide a Markov Chain to rapid convergence.

First, we formulate the problem as Bayesian inference using generative models allowing for a class of shape transformations, see section (2). In section (3), we relate this to the free energy function for the EM algorithm [8] and, thereby, establish a connection to the free energy function used in softassign [5].

Secondly, we define a set of *informative features*, which observe two key properties: *invariant/semi-invariant* and *representative*, to shape transformations such as scaling, rotation, and certain non-rigid transformations, see sections (4.1,4.2). Shape contexts [3] are examples of informative features.

Thirdly, the generative model and informative features are combined in the EM free energy framework, see section (4.3,4.4). The informative features are used as approximations, similar to the proposals in DDMCMC [13], which guide the algorithm to activate the generative models and achieve rapid convergence. Alternatively, one can think of the informative features as providing approximations to the true probabilities distributions, similar to the mean field and Bethe-Kikuchi approximations used by Rangarajan *et al* [5],[9].

We tested our algorithm on a variety of binary and real images and obtained very good performance, see section (6). The algorithms was extensively tested on binary datasets where its performance could be compared to existing algorithms. But we also give results on real images for recognition and detection.

2 Problem Definition

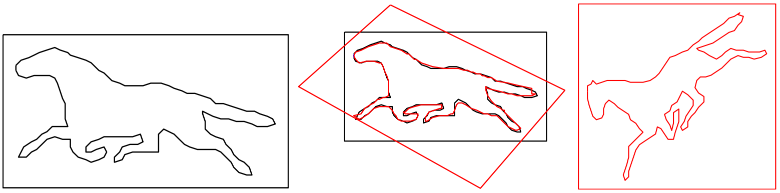
2.1 Shape Representation

The task of shape matching is to match two arbitrary shapes, X and Y, and to measure the similarity (metric) between them. Following Grenander's pattern theory [6], we can define shape similarity in terms of the transformation F that takes one shape to the other, see Fig. 1. In this paper we allow two types of transformation: (i) a global affine transformation, and (ii) a local small and smooth non-rigid transformation.

We assume that each shape is represented by a set of points which are either *sparse* or *connected* (the choice will depend on the form of the input data).

For the **sparse point representation**, we denote the target and source shape respectively by:

$$X = \{(\mathbf{x}_i) : i = 1, ..., M\}, \ and \ Y = \{(\mathbf{y}_a) : a = 1, ..., N\}.$$



(a) Target shape X (b) Transformation F (c) Source shape Y

Fig. 1. Illustration of a shape matching case in which a source shape Y is matched with a target shape X through a transformation function F.

This representation will be used if we match a shape to the edge map of an image.

For the **connected point representation**, we denote the target and source shape respectively by:

$$X = \{(\mathbf{x}(s)) : s \in [0,1]\},\ and\ \ Y = \{(\mathbf{y}(t) : t \in [0,1]\},$$

where s and t are normalized arc-length distances. This model is used for matching shapes to silhouettes. (The extension to multiple contours is straightforward.)

2.2 The Probability Models

We assume a shape X is generated by a shape Y by a transformation $F = (A, \mathbf{f})$ where A is an affine transformation, and $\mathbf{f}$ denotes an non-rigid local transformation (in thin-plate-splines (TPS) [4], the two transformations are combined, but we separate them here for clarity). For any point $\mathbf{y}_a$ on Y, let $v_a \in \{0..M\}$ be the correspondence variable to points in X. For example, $v_a = 4$ means that point $\mathbf{y}_a$ on Y corresponds to point $\mathbf{x}_4$ on X. If $v_a = 0$, then $\mathbf{y}_a$ is unmatched. We define $V = (v_a, a = 1..N)$. The generative model is written as

$$p(X|Y, V, (A, \mathbf{f})) \propto exp\{-E_D(X, Y, V, (A, \mathbf{f}))\},$$

where

$$E_D(X, Y, V, (A, \mathbf{f})) = \sum_a (1 - \delta(v_a))||\mathbf{x}_{v_a} - A\mathbf{y}_a - \mathbf{f}(\mathbf{y}_a)||^2/\sigma^2. \qquad (1)$$

and $(1 - \delta(v_a))$ is used to discount unmatched points (where $v_a = 0$). There is a prior probability $p(V)$ on the matches which pays a penalty for unmatched points. Therefore,

$$p(X, V|Y, V, (A, \mathbf{f})) \propto exp\{-E_T(X, Y, V, (A, \mathbf{f}))\},$$

where $E_T(X, Y, V, (A, \mathbf{f})) = E_D(X, Y, V, (A, \mathbf{f})) - logp(V)$.

The affine transformation A is decomposed [1] as

$$A = \begin{pmatrix} S_x & 0 \\ 0 & S_y \end{pmatrix} \begin{pmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{pmatrix} \begin{pmatrix} 1 & k \\ 0 & 1 \end{pmatrix}.$$

where θ is the rotation angle, S_x and S_y denote scaling, and k is shearing. The prior on A is given by $p(A) \propto exp\{-E_A(A)\}$ where $E_A(A) = E_{rotation}(\theta) + E_{scaling}(S_x, S_y) + E_{shearing}(s)$.

The prior on the non-rigid transformation $\mathbf{f}$ is given by

$$p(\mathbf{f}) \propto exp\{-E_f(\mathbf{f})\},\ and\ \ E_f(\mathbf{f}) = \lambda \int \sum_{m=0}^{\infty} c_m (D^m \mathbf{f})^2 d\mathbf{y},$$

The $\{c_m\}$ are set to be $\sigma^{2m}/(m)2^m$ (Yuille and Grzywacz [15]). This enforces a probabilistic bias for the transformations to be small (the $c_0 = 1$ term) and smooth (the remaining terms $\{c_i : i \geq 1\}$. It can be shown [15], that $\mathbf{f}$ is of the form $\mathbf{f}(x) = \sum_i \alpha_i G(x - x_i)$ where $G(x)$ is the Green's function of the differential operator. We use the Gaussian kernel for $\mathbf{f}$ in this paper (alternative kernels such as TPS give similar results).

The generative model and the prior probabilities determine a similarity measure:

$$D(X||Y) = -\log p(X|Y) = -\log \int \sum_{V} p(X, V, (A, \mathbf{f})|Y) dA d\mathbf{f}. \tag{2}$$

Unfortunately evaluating eqn. (2) requires integrating out $(A, \mathbf{f})$ and summing out V. Both stages are computationally very expensive. Our strategy is to approximate the sum over the V, by using the informative features described in section (4). We then approximate the integral over $(A, \mathbf{f})$ by the modes of $p(A, \mathbf{f}|X, Y)$ (similar to a saddle point approximation). Therefore we seek to find the $(A, \mathbf{f})^*$ that best represent the distribution:

$$\int \sum_{V} p(X, V, (A, \mathbf{f})|Y) dA d\mathbf{f} \sim Par(p(X, (A, \mathbf{f})^*|Y)) \tag{3}$$

where Par is a Parzen window. Our experiments show that the integral is almost always dominated by $(A, \mathbf{f})^*$. Therefore, we approximate the similarity measure by:

$$D_{Appox}(X||Y) = -\log \sum_{V} p(X, V, (A, \mathbf{f})^*|Y), \tag{4}$$

where

$$\begin{aligned}(A, \mathbf{f})^* &= \arg\max_{(A,\mathbf{f})} \sum_{V} p(X, V, (A, \mathbf{f})|Y) \\ &= \arg\min_{(A,\mathbf{f})} -\log \sum_{V} p(X, V|Y, V, (A, \mathbf{f})) p(A) p(\mathbf{f}). \end{aligned} \tag{5}$$

In rare cases, we will require the sum over several models. For example, three modes $((A, \mathbf{f})^*, (A, \mathbf{f})^*_1, (A, \mathbf{f})^*_2))$ are required when matching two equal lateral triangles, see Fig. (2).

Note that this similarity measure is not symmetric between X and Y. But in practice, we found that it was approximately symmetric unless one shape was significantly larger than the other (because of how the A scales the measure). To avoid this problem, we can compute $D(X||Y) + D(Y||X)$. The recognition aspect of the algorithm can be naturally extended from the similarity measure for the two shapes.

3 The EM Free Energy

Computing $(A, \mathbf{f})^*$ in equation (5) requires us to sum out the hidden variable V. This fits the framework of the EM algorithm. It can be shown [8] that estimating $(A, \mathbf{f})^*$ in eqn. (5) is equivalent to minimizing the EM free energy function:

$$\begin{aligned} E(\hat{p}, (A, \mathbf{f})) &= -\sum_V \hat{p}(V) \log p(X, V|Y, (A, \mathbf{f})) - \log p(A, \mathbf{f}) + \sum_V \hat{p}(V) \log \hat{p}(V) \\ &= \sum_V \hat{p}(V) E_T(X, Y, V, (A, \mathbf{f})) + E_A(A) + E_f(\mathbf{f}) + \sum_V \hat{p}(V) \log \hat{p}(V). \end{aligned} \tag{6}$$

The EM free energy is minimized when $\hat{p}(V) = p(V|X, Y, A, \mathbf{f})$. The EM algorithm consists of two steps: (I) The E-step minimizes $E(\hat{p}, (A, \mathbf{f}))$ with respect to $\hat{p}(V)$ keeping $(A, \mathbf{f})$ fixed, (II) The M-step minimizes $E(\hat{p}, (A, \mathbf{f}))$ with respect to $(A, \mathbf{f})$ with $\hat{p}(V)$ fixed. But an advantage of the EM free energy is that any algorithm which decreases the free energy is guaranteed to converge to, at worst, a local minima [8]. Therefore we do not need to restrict ourselves to the standard E-step and M-step.

Chui and Rangarajan's free energy [5],

$$E(M, f) = \sum_{i=1}^{N} \sum_{a=1}^{N} m_{ai} ||x_i - f(v_a)||^2 + \lambda ||Lf||^2 + T \sum_{i=1}^{N} \sum_{a=1}^{K} m_{ai} \log m_{ai} - \zeta \sum_{i=1}^{N} \sum_{a=1}^{K} m_{ai} \tag{7}$$

can be obtained as a *mean field approximation* to the EM free energy. This requires assuming that $\hat{p}(V)$ can be approximated by a factorizable distribution $\prod_a P(v_a)$. The soft-assign variables $m_{ai} \in [0, 1]$ are related to $\hat{p}(V)$ by $m_{ai} = \hat{P}(v_a = i)$. An alternative approximation to the EM free energy can be done by using the Bethe-Kikuchi free energy [9].

Like Rangarajan *et al* [5,9] we will need to approximate $\hat{p}(V)$ in order to make the EM algorithm tractible. Our approximations will be motivated by informative features, see section (4), which will give a link to shape contexts [3] and feature-based algorithms.

4 Implementing the EM Algorithm

In this section, we introduce informative features and describe the implementation of the algorithm.

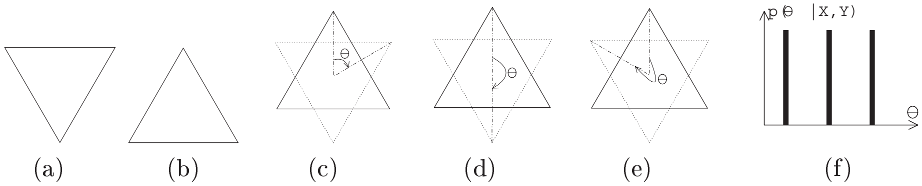


Fig. 2. The distribution $p(\theta|X, Y)$, shown in (f), has three modes for a target shape X, shown in (a), and a source shape Y, shown in (b). (c), (d), and (e) respectively display the three possible values for the θ.

4.1 Computing the Initial State

The EM algorithm is only guaranteed to converge to a local minima of the free energy. Thus, it is critical for the EM algorithm to start with the "right" initial state. Our preliminary experiments in shape matching suggested that the probability distribution for $(A, \mathbf{f})$ is strongly peaked and the probability mass is concentrated in small areas around $\{(A, \mathbf{f})^*, (A, \mathbf{f})^*_2, (A, \mathbf{f})^*_3, ...\}$. Hence if we can make good initial estimates of $(A, \mathbf{f})$, then EM has a good chance of converging to the global optimum.

The rotation angle θ is usually the most important part of $(A, \mathbf{f})$ to be estimated. (See Fig.2 for an example where there are three equally likely choices for θ.) It would be best to get the initial estimate of θ from $p(\theta|X, Y)$, but this requires integrating out variables which is computationally too expensive. Instead, we seek to approximate $p(\theta|X, Y)$ (similar to the Hough Transform [2]) by an *informative feature distribution* $p_{IF}(\theta|X, Y)$:

$$p(\theta|X, Y) \approx p_{IF}(\theta|X, Y) = \sum_i \sum_a q(\phi(\mathbf{x}_i), \phi(\mathbf{y}_a))\delta(\theta - \theta(a, i, X, Y)), \tag{8}$$

where $\phi(\mathbf{x}_i)$ and $\phi(\mathbf{y}_a)$ are *informative features* for point $\mathbf{x}_i$ and y_a respectively, $q(\mathbf{x}_i, \mathbf{y}_a)$ is a *similarity measure* between the features, and $\theta(X, Y, a, i)$ is the angle if the ith point on X is matched with ath point on Y.

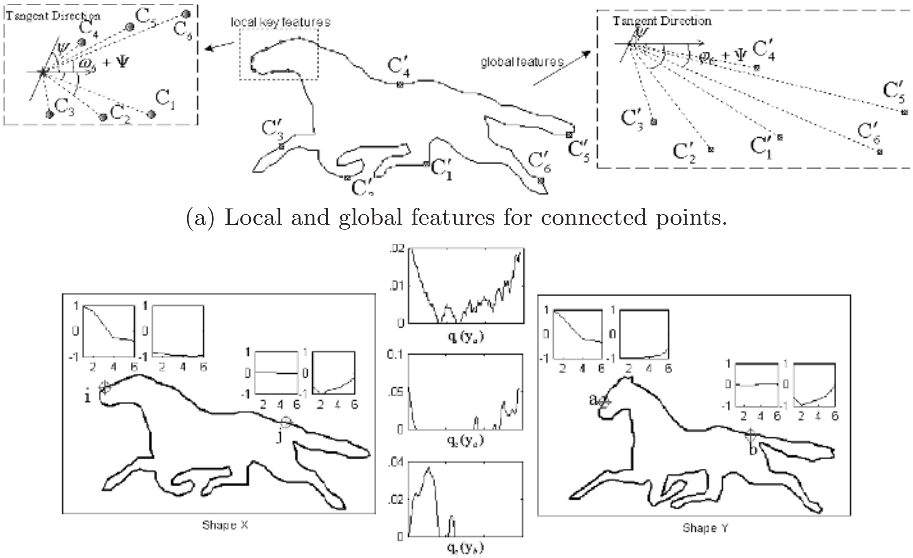
Next, we describe how to design the informative features $\phi(\mathbf{x}_i)$ and the similarity measures $q(\phi(\mathbf{x}_i), \phi(\mathbf{y}_a))$.

4.2 Designing the Informative Features

The *informative features* are used to make computationally feasible approximations to the true probability distributions. They should observe two key properties to have

$$\int p(\theta|X, Y, (A_{\_\theta}, \mathbf{f}))p(A_{\_\theta})p(\mathbf{f})dA_{\_\theta}d\mathbf{f} \approx p(\theta|\phi(X), \phi(Y))$$

(I) They should be "invariant" as possible to the transformations. Ideally $p(\theta|\phi(X), \phi(Y), (A_{\_\theta}, f)) = p(\theta|\phi(X), \phi(Y))$.



(a) Local and global features for connected points.

(b) Similarity measure on the features

Fig. 3. Features and the similarity measure of the features. a) Illustrates how the local and global features are measured for connected points. In b)., the features of two points in shape X and Y are displayed. The top figure in the middle of b). shows similarities between point a in Y w.r.t. all points in X using the shape context feature. The other two figures in the middle of b). are the similarities between points a and b in Y w.r.t. all points in X respectively. As we can see, similarities by features defined in this paper for connected points have lower entropy than those by shape contexts.

(II) They should be "representative". For example, we would ideally have

$$\int p(\theta|X, Y, (A_{\_\theta}, \mathbf{f}))p(A_{\_\theta})p(\mathbf{f})dA_{\_\theta}d\mathbf{f} = \int p(\theta|\phi(X), \phi(Y), (A_{\_\theta}, \mathbf{f}))p(A_{\_\theta})p(\mathbf{f})dA_{\_\theta}d\mathbf{f}$$

where $A_{\_\theta}$ is the components of A except for θ and $\phi(X), \phi(Y)$ are the feature vectors for all points in both images.

The two properties for imformative features are also used to approximate distribution of other variables, for example, $p(V|X, Y)$, which requires us to integrate out $(A, \mathbf{f})$ and can be approximated by $p_{IF}(V|\phi(X), \phi(Y)) = \prod_a q(\phi(\mathbf{x}_{v_a}), \phi(\mathbf{y}_a))$.

In this paper we select the features $\phi(.)$ and measures $q(.,.)$ so as to obtain P_{IF} with *low-entropy*. This is a natural choice because it implies that the features have low matching ambiguity. We can evaluate this *low-entropy* criteria over our dataset for different choices of features and measures, see figure (3).

A better criteria, though harder to implement, is to select the features and measures which maximize the conditional Kullback-Leibler divergence evaluated over the distribution $p(X, Y)$ of problem instances:

$$\sum_{X,Y} p(X,Y) \sum_{(A,\mathbf{f})} p(V,(A,\mathbf{f})|X,Y) \log \frac{p(V,(A,\mathbf{f})|X,Y)}{p_{IF}(V,(A,\mathbf{f})|\phi(X),\phi(Y))}, \tag{9}$$

but full evaluation of this criterion is a task for future work.

We used the low-entropy criterion to devise different sets of features for the two cases, shapes of *connected* points representation and shapes of *sparse* points representation.

Case I: The Connected Point Representation

We use **local** and **global** features illustrated by Fig.3. The local features at a point $\mathbf{x}(s_i)$ with tangent ψ_i are defined as follows. Choose six points on the curve by $(\mathbf{x}(s_i-3ds), \mathbf{x}(s_i-2ds), \mathbf{x}(s_i-ds), \mathbf{x}(s_i+ds), \mathbf{x}(s_i+2ds), \mathbf{x}(s_i+3ds))$, where ds is a (small) constant. The angles of these positions w.r.t. point $\mathbf{x}_i$ are $\psi_i+\omega_j, j = 1..6)$. *The local features are* $h_l(\mathbf{x}_i) = (\omega_j, j = 1..6)$. The global features are selected in a similar way. We choose six points near $\mathbf{x}(s_i)$, with tangent ψ_i, to be $(\mathbf{x}(s_i-3\Delta s), \mathbf{x}(s_i-2\Delta s), \mathbf{x}(s_i-\Delta s), \mathbf{x}(s_i+\Delta s), \mathbf{x}(s_i+2\Delta s), \mathbf{x}(s_i+3\Delta s))$, where Δs is a (large) constant, with angles $\psi_i+\varphi_j : j = 1, ..., 6$. *The global features are* $h_g(\mathbf{x}_i) = (\varphi_j, j = 1..6)$. Observe that the features $\phi = (h_l, h_g)$ are invariant to rotations in the image plabe and also, to some extent, to local transformations.

In Fig. (3).b., for display purposes we plot sinusoids $(sin(h_l), sin(h_g))$ for two points on the X and two points on the Y. Observe the similarity between these features on the corresponding points.

The similarity measure between the two points is defined to be:

$$q_c(\phi(\mathbf{x}_i), \phi(\mathbf{y}_a)) = 1 - c_1(\sum_{j=1}^{6} D_{angle}(\omega_j(x_i) - \omega_j(y_a)) + \sum_{j=1}^{6} D_{angle}(\varphi_j(x_i) - \varphi_j(y_a))),$$

where $D_{angle}(\omega_j(x_i) - \omega_j(y_a))$ is the minimal angle from $\omega_j(x_i)$ to $\omega_j(y_a)$, and c_1 is a normalization constant. The second and the third row in the middle of Fig. (3).b. respectively plot the vector $q_c(\mathbf{y}) = [q_c(\phi(\mathbf{x}_i), \phi(\mathbf{y})), i = 1..M]$ as a function of i for points $\mathbf{y}_a$ and $\mathbf{y}_b$ on Y.

Case II: The Sparse Point Representation

In this case, we also use **local** and **global** features. To obtain the local feature for point $\mathbf{x}_i$, we draw a circle with a (small) radius r and collect all the points that fall into the circle. The relative angles of these points w.r.t. $\mathbf{x}_i$ and $\mathbf{x}_i$'s tangent angle are computed. The histogram of these angles is then used as the local feature, H_l.

The global feature for the sparse points is computed by shape contexts [3]. We denote it by H_g and the features become $\phi = (H_l, H_g)$.

The feature similarity between two points $\mathbf{x}_i$ and y_a is measured by the χ^2 distance:

$$q_s(\phi(\mathbf{x}_i), \phi(\mathbf{y}_a)) = 1 - c_2(\chi^2(H_l(\mathbf{x}_i), H_l(\mathbf{y}_a)) + \chi^2(H_g(\mathbf{x}_i), H_g(\mathbf{y}_a))).$$

The first row in the middle of Fig. (3).b. plots the vector

$$q_s(\mathbf{y}_a) = [q_s(\phi(\mathbf{x}_i), \phi(\mathbf{y}_a)), i = 1..M]$$

as a function of i for a point $\mathbf{y}_a$ on Y.

The advantage of the sparse point representation is that it is very general and does not require a procedure to group points into contours. But for this very reason, the features and measures have higher entropy than those for the connected point representation. In particular, the global nature of the shape context features [3] means that these features and measures tend to have high entropy, see the Fig. (3)b., particularly shape context features are also of unnecessarily high dimension – consisting of 2D histograms with 60 bins – and better results, in terms of entropy, can be obtained with lower dimensional features.

4.3 The E Step: Approximating $\hat{p}(V)$

We can obtain an approximation $p_{IF}(\theta|X,Y)$ to $p(\theta|X,Y)$, see equation (8), using the informative features and similarity measures described in the previous section. We select each peak in $p_{IF}(\theta|X,Y)$ as an initial condition $\theta_{initial}$ for θ. The same approach is used to estimate the other variables in A and $\mathbf{f}$ from $p(V|X,Y,\theta_{initial})$. We use similar informative features to those described in the previous section except that we replace ψ by $\theta_{initial}$

$$h'_l = (\omega'_j, j = 1..6) = (\alpha_j - \theta_{initial}, j = 1..6),$$

and

$$h'_g = (\varphi'_j, j = 1..6) = (\beta_j - \theta_{initial}, j = 1..6). \tag{10}$$

We also augment the similarity measure by including the scaled relative position of point $\mathbf{x}_i$ to the center of the shape $\bar{\mathbf{x}} = \frac{1}{M}\sum_i \mathbf{x}_i$:

$$q'_c(\phi(\mathbf{x}_i), \phi(\mathbf{y}_a)) = 1 - c'_1 \sum_{j=1}^{6} [D_{angle}(\omega'_j(x_i) - \omega'_j(y_a)) + D_{angle}(\varphi'_j(x_i) - \varphi'_j(y_a)))] \\ - c'_2 ||\mathbf{x}_i - \bar{\mathbf{x}}, \mathbf{y}_a - \bar{\mathbf{y}}||^2.$$

Thus, we have the following approximation:

$$p(V|X,Y,\theta) \approx p_{IF}(V|X,Y,\theta) = \prod_a p_{if}(v_a|y_a,X,Y,\theta). \tag{11}$$

where

$$p_{if}(v_a = i|y_a,X,Y,\theta) \approx \frac{q'_c(\phi(\mathbf{x}_i), \phi(\mathbf{y}_a))}{\sum_{j=0}^{M} q'_c(\phi(\mathbf{x}_j), \phi(\mathbf{y}_a))}.$$

After the first iteration, we update the features and feature similarity measure by $q'_c(\phi(\mathbf{x}_i), \phi((A+\mathbf{f})(\mathbf{y}_a)))$ and use them to approximate $p(V|X,Y,(A,\mathbf{f}))$ as in eqn (11).

4.4 The M Step: Estimating A and f

Once we have an approximation to $\hat{p}(V)$, we then need to estimate $(A,\mathbf{f})$ according to eqn. (6). We expand $E(\hat{p},(A,\mathbf{f}))$ as a Taylor series in $A, \mathbf{f}$ keeping the second order terms and then estimate $(A,\mathbf{f})$ by least squares.

5 Summary of the Algorithm

Our algorithm is performed by an approximation to the EM algorithm and it proceeds as follows:

1. Given a target shape X and a source shape Y, it computes their informative features described in section 4.2 and uses $P_{IF}(\theta|X,Y)$ (equation (8)) to obtain several possible rotation angles $\theta_{initial}$s.
2. For each rotation angle $\theta_{initial}$, we obtain a new shape Y' by rotating it for $\theta_{initial}$.
3. Update features for shape Y' and estimate $p(V|X,Y,\theta)$ by $P_{IF}(V|XY,\theta)$ as eqn. (11).
4. Estimate $(A,\mathbf{f})$ from the EM equation by least-squares method.
5. Obtain the new shape Y' by the transformation function $A\mathbf{y}+\mathbf{f}(Y)$. Repeat step 3 for 4 iterations.
6. Compute the similarity measure and keep the best $(A,\mathbf{f})^*$, among all the initial $\theta_{initial}$s and compute the metric according to eqns. (3) and (2). (We can also combine the results from several starting points to approximate eqn. 2. In practice, we found there is not much difference except for special cases like the equal lateral triangle.)

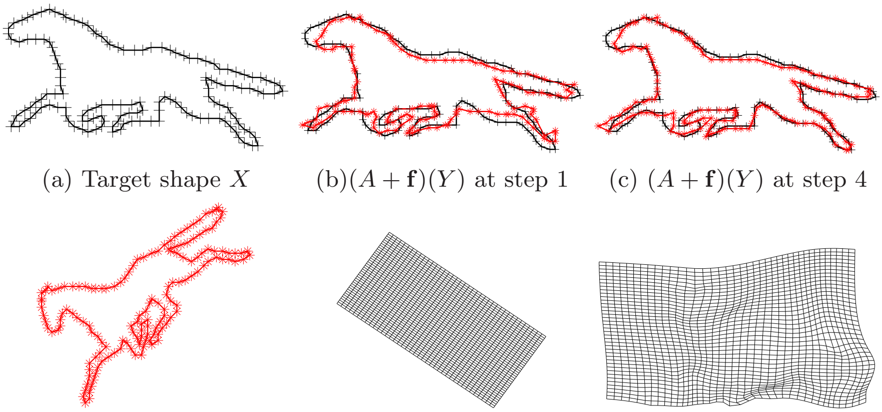
The algorithm runs at 0.2 seconds for matching X and Y of around 100 points. Note that our method does not need the target shape X and the source shape Y to have the same or nearly the same number of points, which is a key requirement for many matching algorithms.

6 Experiments

We tested our algorithm on a variety of data sets and some results are reported in this section. Fig.4 shows the running example where the source shape Y in (d) is matched with the target shape X. Fig.4.e and .f show the transformation A^* and $\mathbf{f}^*$ estimated.

6.1 MPEG7 Shape Database

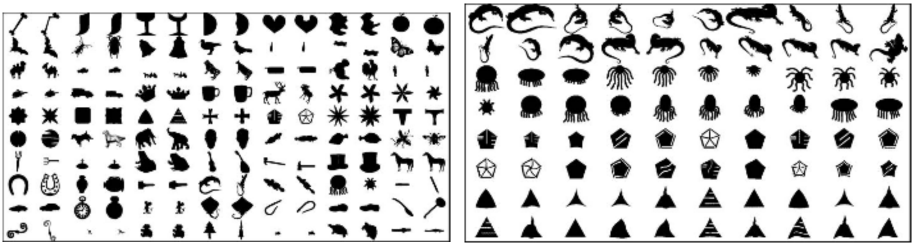
We first tested our algorithm on the MPEG7 CE-Shape-1 [7] which consists of 70 types of objects each of which has 20 different silhouette images (i.e. a total of 1400 silhouettes). Since the input images are binarized, we can extract contours and use the connected point representation. Fig.5.a displays 2 images for each type. The task is to do retrieval and the recognition rate is measured by "Bull's eye" [7]. For every image in the database, we match it with every other image and keep the 40 best matched candidates. For each one of the other 19 of the same type, if it is in the selected 40 best matches, it is considered as a success. Observe that the silhouettes also include mirror transformations which our algorithm can take into account because the informative features are computed based on relative angles. The recognition rates for different algorithms are shown in table 1 [10] which shows that our algorithms outperforms the alternatives. The speed is in the same range as those of shape contexts [3] and curve edit distance [10].



(a) Target shape X (b)$(A + \mathbf{f})(Y)$ at step 1 (c) $(A + \mathbf{f})(Y)$ at step 4

(d) Source shape Y (e) A^* (f) $\mathbf{f}^*$

Fig. 4. Shape matching of the running example.



(a) Some typical images in MPEG7 CE-Shape-1 (b) The four types with the lowest rates

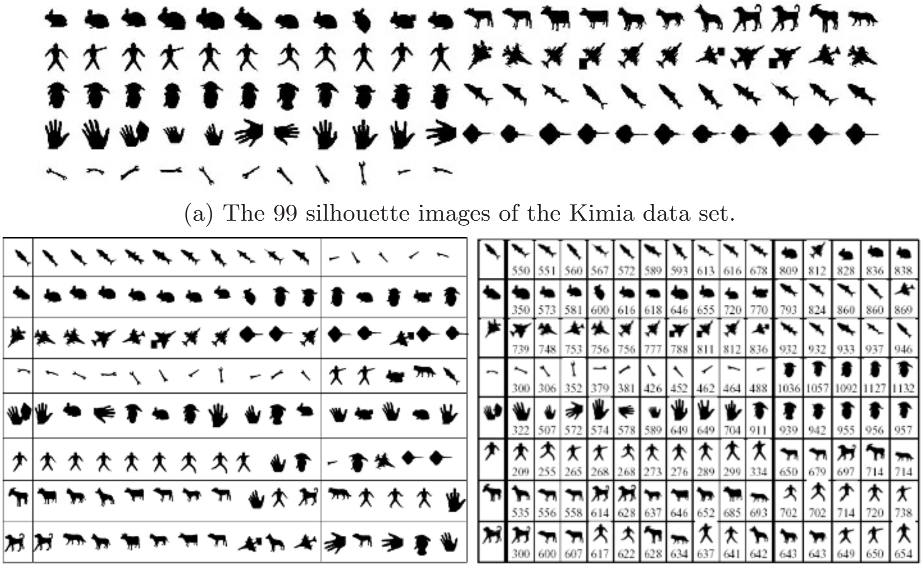
Fig. 5. Matching as image retrieval for the MPEG7 CE-Shape-1.

Table 1. The retrieval rates of different algorithms for the MPEG7 CE-Shape-1. Results by the other algorithms are from Sebastian et al. [10].

Algorithm	CSS	Visual Parts	Shape Contexts	Curve Edit Distance	Our Method
Recognition Rate	75.44%	76.45%	76.51%[3]	78.17% [10]	80.03%

6.2 The Kimia Data Set

We then tested the identical algorithm (i.e. connected point representation and same algorithm parameters) on the Kimia data set of 99 shapes [11], which are shown in Fig.6.a. For each shape, the 10 best matches are picked since there are 10 other images in the same category. Table 2 shows the numbers of correct matches. Our method performs similarly to Shock Edit [11] for the top 7 matches, but is worse for the top 8 to 10. Shape context performs less well than both algorithms on this task. Fig.6.b. displays the fifteen top matches for some shapes. Our relative failure, compared with Shock Edit, is due to the transformations which occur in the dataste, see the 8-10^{th} examples for each object in figure (6), and which require more sophisticated representations and transformations than those used in this paper.



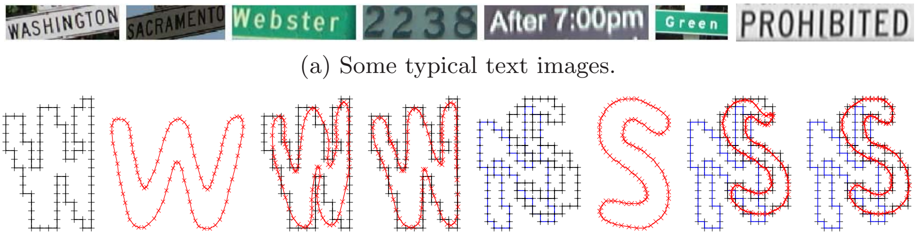
(a) The 99 silhouette images of the Kimia data set.

(b) Some matching results by our method (c) Some matching results by Shock Edit

Fig. 6. The Kimia data set of 99 shapes and some matching results.

Table 2. Numbers of matched shapes by different algorithms. Results by the other algorithms are due to Sebastian et al. [11].

Algorithm	Top 1	Top 2	Top 3	Top 4	Top 5	Top 6	Top 7	Top 8	Top 9	Top 10
Shock Edit	99	99	99	98	98	97	96	95	93	82
Our Method	99	97	99	98	96	96	94	83	75	48
Shape Contexts	97	91	88	85	84	77	75	66	56	37



(a) Some typical text images.

(b) X_1 (c) Y_1 (d) (e) (f) X_2 (g) Y_2 (h) (i)

Fig. 7. Results on some text images. (e) and (i) display the matching. We purposely put two shapes together and find that the algorithm is robust in this case.

6.3 Text Image Matching

The algorithm was also tested on real images of text in which binarization was performed followed by boundary extraction. Some examples are shown in Fig.7. Similar results can be obtained by matching the model to edges in the image. Further tests on this dataset are ongoing.

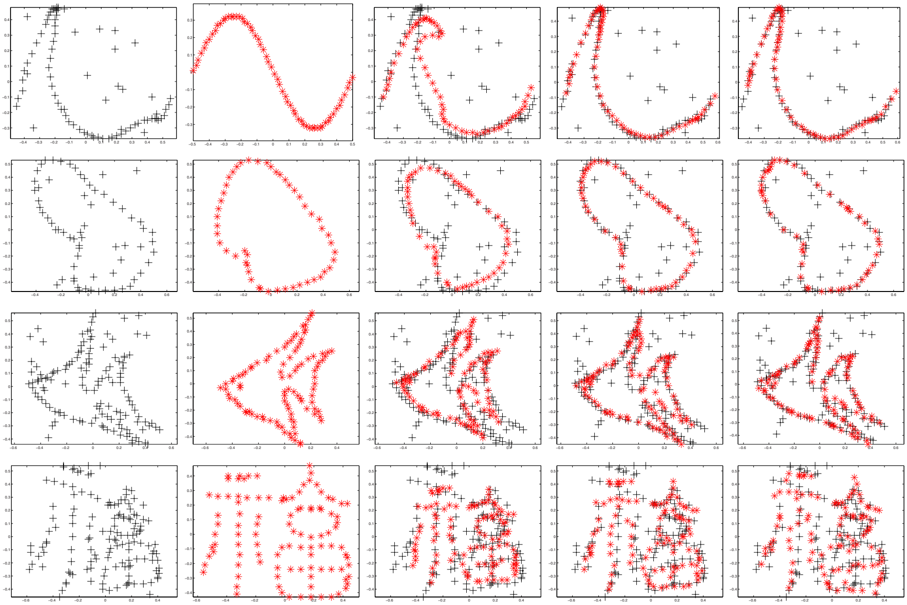

Target shape X Source shape Y Step 1 Step 5 Step 10

Fig. 8. Some results on Chui and Rangarajan data set.

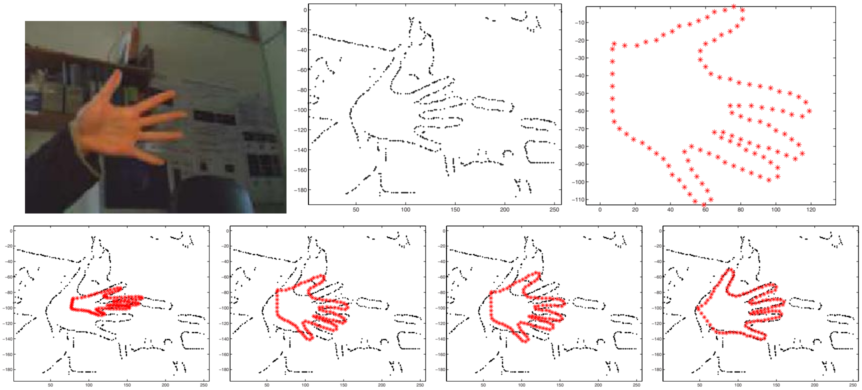


Fig. 9. Result on a hand image.

6.4 Chui and Rangarajan

To test our algorithm as a shape registration method, we also tried the data set used by Chui and Rangarajan [5]. We used the sparse point representation in this case. The algorithm runs for 10 steps and some results are shown in Fig.8. The quality of our results are similar to those reported in [5]. But our algorithm runs an estimated 20 times faster.

6.5 A Detection Task

Our algorithm can also be used for object detection where, unlike recognition, we do not know where the object is in the image. To illustrate this, we tested our algorithm on a hand image used in [14]. Edge points were extracted to act as the target shape and the source image was a hand represented by sparse points. The result is shown in Fig.9.

7 Discussion

This paper introduced a criterion for shape similarity and an algorithm for computing it. Our approach helps show relations between softassign [5] and shape contexts [3]. We formulated shape similarity by a generative model and used a modified variant of the EM algorithm for inference. A key element is the use of informative features to guide the algorithm to rapid and correct solutions. We illustrated our approach on datasets of binary and real images, and gave comparison to other methods. Our algorithm runs at speeds which are comparable to alternatives and is faster than others by orders of magnitude.

Our work is currently limited by the types of representations we used and the transformations we allow. For example, it would give poor results for shape composed of parts that can deform independently (e.g. human figures). For such objects, we would need representations based on symmetry axes such as skeletons [10] and parts [16]. Our current research is to extend our method to deal with such objects and to enable the algorithm to use input features other than edge maps and binary segmentations.

References

1. S. Abbasi and F. Mokhtarian, "Robustness of Shape Similarity Retrieval under Affine", *Proc. of Challenge of Image Retrieval*, 1999
2. D. H. Ballard, "Generalizing the Hough Transform to Detect Arbitrary Shapes", *Pattern Recognition*, vol. 13, no. 2, 1981.
3. S. Belongie, J. Malik, and J. Puzicha, "Shape Matching and Object Recognition Using Shape Contexts", *IEEE Trans. on PAMI*, vol. 24, no. 24, 2002.
4. F. L. Bookstein, "Principal Warps: Thin-Plate Splines and the Decomposition of Deformations", *IEEE Trans. on PAMI*, vol. 11, no. 6, 1989.
5. H. Chui and A. Rangarajan, "A New Point Matching Algorithm for Non-rigid Registration", *Computer Vision and Image Understanding*, March, 2003.
6. U. Grenander, "General Pattern Theory: A Mathematical Study of Regular Structures", *Oxford*, 1994.
7. L. J. Latechi, R. Lakamper, and U. Eckhardt, "Shape Descriptors for Non-rigid Shapes with a Single Closed Contour", *Proc. of CVPR*, 2000.
8. R. Neal and G. E. Hinton, "A View Of The Em Algorithm That Justifies Incremental, Sparse, And Other Variants", *Learning in Graphical Models*, 1998.
9. A. Rangarajan, J.M. Coughlan, A.L. Yuille, "A Bayesian Network for Relational Shape Matching", *Proc. of ICCV*, Nice, France, 2003.

10. T. B. Sebastian, P. N. Klein, and B. B. Kimia, "On Aligning Curves", *IEEE Trans. on PAMI*, vol. 25, no. 1, 2003.
11. T. B. Sebastian, P. N. Klein, and B. B. Kimia, "Recognition of Shapes by Editing their Shock Graphs", *accepted by IEEE Trans. on PAMI*, 2003.
12. A. Thayananthan, B. Stenger, P.H.S. Torr, and R. Cipolla, "Shape Context and Chamfer Matching in Cluttered Scenes", *CVPR*, 2003.
13. Z. Tu, X. Chen, A. Yuille, and S.C. Zhu, "Image Parsing: Unifying Segmentation, Detection and Recognition", *Proc. of ICCV*, Nice, France, 2003.
14. R. C. Veltkamp and M. Hagedoorn, "State of the Art in Shape Matching", Technical Report UU-CS-1999-27, Utrecht, 1999.
15. A. L. Yuille and N. M. Grzywacz, "A Computational Theory for the Perception of Coherent Visual Motion", *Nature*, vo. 333, no. 6168, 1988.
16. S. C. Zhu and A. L. Yuille, "FORMS: A Flexible Object Recognition and Modeling System", *IJCV*, vol.20, no.3, 1996.