Learning Compositional Shape Models of Multiple Distance Metrics by Information Projection

Ping Luo, Liang Lin, and Xiaobai Liu

***Abstract*— This paper presents a novel compositional contour-based shape model by incorporating multiple distance metrics to account for varying shape distortions or deformations. Our approach contains two key steps: 1) contour feature generation and 2) generative model pursuit. For each category, we first densely sample an ensemble of local prototype contour segments from a few positive shape examples and describe each segment using three different types of distance metrics. These metrics are diverse and complementary with each other to capture various shape deformations. We regard the parameterized contour segment plus an additive residual ϵ as a basic subspace, namely, ϵ-ball, in the sense that it represents local shape variance under the certain distance metric. Using these ϵ-balls as features, we then propose a generative learning algorithm to pursue the compositional shape model, which greedily selects the most representative features under the information projection principle. In experiments, we evaluate our model on several public challenging data sets, and demonstrate that the integration of multiple shape distance metrics is capable of dealing various shape deformations, articulations, and background clutter, hence boosting system performance.**

***Index Terms*— Compositional model, information projection, object detection, shape analysis.**

I. Introduction

A. Motivation and Overview

SHAPE detection is a great challenge, especially when various shape deformations are presented, such as scale, rotation, articulation, and distortion. Numerous methods have been proposed for capturing shape deformations on binary images such as [1]–[3] and achieved impressive results. However, these techniques cannot be simply applied to natural images because of the following two difficulties. First, objects in real-world images often appear in different views and poses, leading to large shape variance, e.g., distortions and deformations. Fig. 1(a) and (b) shows a cross and its instance image. One needs to scale and twist the cross in Fig. 1(a) while seeking its correspondence in Fig. 1(b). Another example is to locate the *Horse shape* [Fig. 1(e)] in Fig. 1(f). Second, object boundaries are usually incomplete due to noises or occlusions. Two exemplars are shown in Fig. 1(c) and (g), and their edge maps are in Fig. 1(d) and (h), respectively, where the boundary of an object is broken into contour segments or occluded.

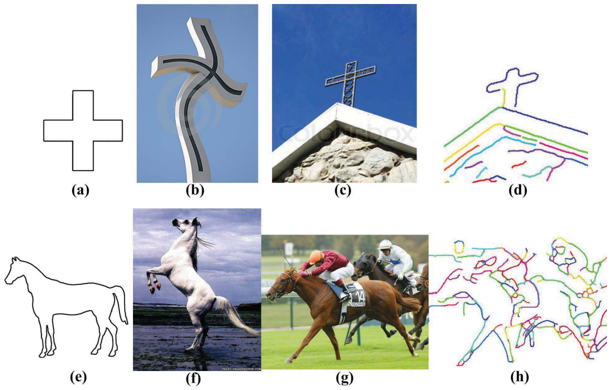


Fig. 1. Illustration of large shape variance. (a) and (e) Two shapes to be matched to four candidate images, i.e., (b), (c), (f), and (g). This task is difficult because shape deformation, occlusion, and background clutter are induced in the images. (d) and (h) Edge maps of (c) and (g), respectively.

In this paper, we present an effective compositional shape model composed of local contour segments, which incorporates multiple types of shape distance metrics to cope with the abovementioned challenges. To make different metrics comparable with each other, we introduce a novel generative learning algorithm that greedily selects the most representative contour-based features to pursue the optimal shape model. Our approach includes two key steps: 1) contour feature generation and 2) generative shape model pursuit.

In the first step, a number of contour segments are extracted from shape examples and represented with different distance metrics. For each shape category, we first collect a small set of typical shape examples, which are chosen to well cover the variances (e.g., different views) of this object category as shown at the left-hand side of Fig. 2(a). Then, for each prototype shape, we extract a number of prototype contour segments with different lengths. As shown in Fig. 2, we further represent each contour segment with three types of shape distance metrics, including procrustes distance [4], articulation distance, and geodesic distance [5]. This is inspired by the classical work of shape analysis [6], which shows that the

Manuscript received October 3, 2014; revised February 4, 2015 and May 6, 2015; accepted May 31, 2015. This work was supported in part by the National Natural Science Foundation of China under Grant 61173082, in part by Guangdong Natural Science Foundation under Grant S2013050014548 and Grant 2014A030313201, in part by the Special Project on Integration of Industry, Education and Research of Guangdong Province under Grant 2012B091000101, and in part by the Program of Guangzhou Zhujiang Star of Science and Technology under Grant 2013J2200067. *(Corresponding author: Liang Lin.)*

P. Luo and L. Lin are with Sun Yat-sen University, Guangzhou 510006, China (e-mail: pluo.lhi@gmail.com; linliang@ieee.org).

X. Liu is with the Department of Statistics, University of California at Los Angeles, Los Angles, CA 90095 USA (e-mail: lxb@ucla.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2015.2440430

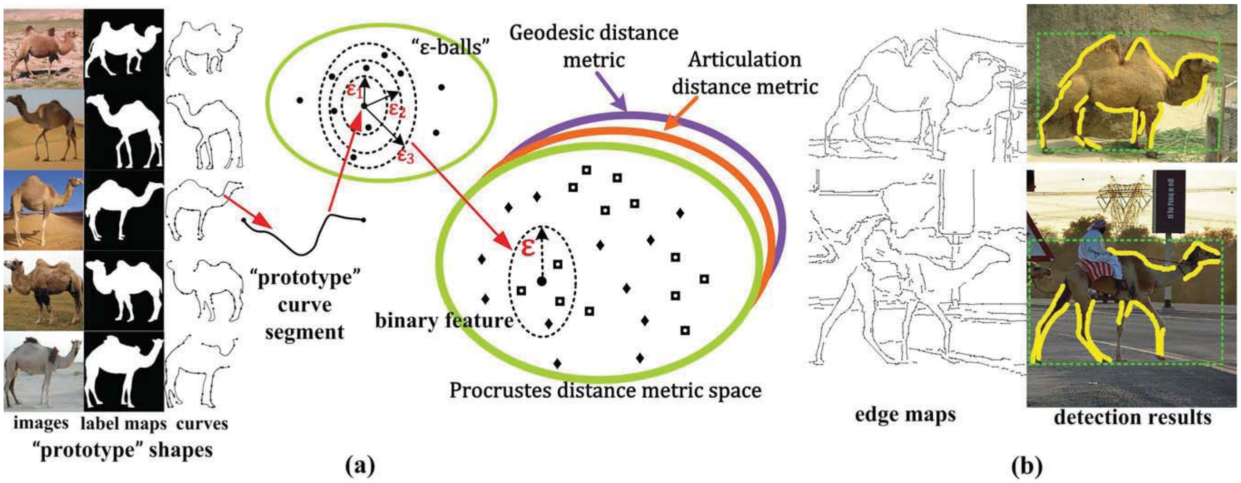


Fig. 2. Illustration of shape modeling and detection with local contour segments. (a) We extract a number of contour segments from typical shape instances for each category. (b) Extracted contour segments are spanned into subspaces, namely, ϵ-balls, by mapping them with different types of distance metrics. These metrics are shown to well capture different types of shape deformations and variants. Then, we build shape models with the ϵ-balls, and some examples of shape detection are also proposed.

arbitrary deformation of a contour can be decomposed into three types: 1) the rigid (affine) transformation; 2) articulation; and 3) distortion (twist). The three distance metrics we employed are shown by [4]–[8] well capturing the abovementioned deformations, respectively. In the rest of this paper, we call these prototype contour segments as proto-seg for simplicity.

Intuitively, each proto-seg can be viewed as a point in the metric space, where this point can be further spanned into a subspace by introducing a statistical fluctuation (i.e., residual) ϵ. As shown in Fig. 2(a), we define the subspace centering at each proto-seg as an ϵ-ball, which represents the shape variance by the proto-seg under a certain metric. Thus, each ϵ-ball can be regarded as a feature or weak classifier that decides whether a shape has a similar local variance compared with it. The step of contour feature generation is capable of discovering intrinsic local shape variance using different shape distance metrics.

In the second step, the proposed compositional shape model is trained by adaptively selecting ϵ-balls as local informative features. In [9], feature selection is usually performed toward a discriminative goal by minimizing the classification errors over a set of labeled negative and positive samples. In contrast, it is also desired to learn generative models [10]–[12] due to the following facts.

1) Generative models are capable of explicitly capturing the data variation and, thus, are more expressive than discriminative models.
2) The models can be effectively visualized (synthesized) for validation using the sampling processes.

In this paper, we propose a learning algorithm for generative models, based on information projection principle [13], to select ϵ-balls according to their information gains. We choose for each category a number of positive samples and a number of reference samples, to build the training set. Intuitively, an ϵ-ball has a higher information gain if its feature statistics is consistent within the training examples and is distinctive from the statistics of reference examples. In the training stage, our learning algorithm starts with an initial shape model and proceeds to pursue a sequence of probability models while gradually minimizing Kullback–Leibler (KL) divergence between the current model and the previous learned model. In the testing stage, given the learned shape model, we adopt the sliding window approach to localize shapes in cluttered edge maps.

To the best of our knowledge, the proposed approach is the first attempt that fuses multiple types of shape distance metrics by pursuing a generative model. Our approach is evaluated on several challenging data sets, including the shape data set with 20 animal classes [14], the ETHZ shape data set [9], the UIUC-People [15], and one subset with 40 categories chosen from the LHI database [16], and compared with other popular methods. The results show that our approach can achieve or advance the state of the art of shape detection.

B. Related Work

We first review the works of shape descriptors and shape metrics, and then discuss the learning-based techniques for constructing shape models.

1) Shape Descriptors and Distance Metrics: A 2-D shape is typically represented by a set of landmarks, each indicating a point with x and y coordinates sampled from shape boundary. This raw representation, however, is sensitive to simple shape deformations, such as affine transformation. To account for these deformations, a variety of shape descriptors and distance metrics have been proposed in the literature.

The shape boundary has been described by the Fourier descriptor [17], skeletons [18], and moment-based features [19]. Also, there are methods exploring the 2-D shape space that assumes as a Riemannian manifold. For example, Klassen *et al.* [20] presented a differential geometric shape representation by employing the direction and curvature on the shape boundary. As the above methods mainly capture global information of shape, the major limitation is that they are sensitive to local changes. To address this problem, some methods decomposed shape representations into local contours [3], as local variances can be better captured. For example, Hoffman and Richards [21] proposed to partition an object shape into parts at negative curvature minima. A tree structure was discussed in [2] to match shapes by recursively decomposing contours.

Shape descriptors are often incorporated with the shape distance metrics [22]–[24], to handle various shape deformations. For example, early works such as the polynomial approximation and its extensions [25] minimized Euclidean distance between two point sets extracted along the boundary. Another representative work is the procrustes distance [4], which infers the scale factor and rotation matrix between two shapes. The above methods were proved to be insensitive to affine transformation. Nevertheless, the inner distance [8] demonstrated its effectiveness in handling shape matching against large articulation deformations. Moreover, the methods [26], [27] based on thin plate spline have been widely used as the nonrigid transformation model for shape matching.

Motivated by the effectiveness of the existing shape distance metrics, we target on utilizing them in a combinatorial and general manner.

2) Shape Model Learning: To solve the category-level shape classification and detection, which is also the focus of this paper, one can learn a shape model for each shape category. These methods often represent shapes as a loose collection of local shape signatures (e.g., small regions, edge pixels, or contour segments) that are described with a certain distance metric. For example, Gu *et al.* [28] separate the object shape into grid and organize the cells of grid into a tree-structure model. The shape band model [29] describes shape as a set of dense discrete points sampled from edges and calculates the orientation of points as features. More methods define shape model based on local contour segments [3], [22], [30]–[33]. These shape models can be trained by employing diverse learning techniques. Shotton *et al.* [34] employ boosting to select contour fragments, and a similar approach is discussed in [30]. SVM-based learning algorithms are also explored on this task, such as [31], [32], and [35]–[37]. Zhu *et al.* [31] further incorporate set-to-set matching into a discriminative framework in which contour words and spatial layout can be determined together to optimize classification performance. Our approach is partially motivated by these methods, but the main distinctions are as follows.

1) We adaptively incorporate multiple distance metrics on local contour segments.
2) The proposed shape model is generative to explain the shape variance rather than only discriminatively selecting features.

The remainder of this paper is organized as follows. We introduce the shape model in Section II and discuss the process of generating ϵ-balls for contour segments in Section II-A. The shape distance metrics and the method for shape detection are introduced in Sections III and IV, respectively. The experimental evaluations are reported in Section V, and this paper is concluded with a summary in Section VI.

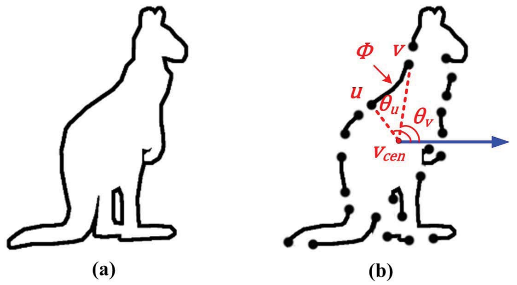


Fig. 3. Contour-segment-based shape representation. (a) Contour (boundary) of shape. (b) Shape (a) using contour segments with different lengths and overlapping. For each contour segment, we encode its spatial configuration with respect to the shape.

II. Shape Modeling

Our method represents a shape $\mathbf{S}$ by a batch of contour segments $\{c\}$, which are randomly sampled from shape boundary. The segments are with varying lengths and allowed to be overlapped with each other. As shown in Fig. 3, we encode each contour segment by its spatial configuration with respect to the shape. Each contour segment c is described by a three-tuple $\{\Phi, \theta_u, \theta_v\}$, where Φ is a set of uniformly sampled points on it, and θ_u and θ_v are the angles of its two end points with respect to the mass center of the shape v_{cen}.

At the beginning, we first extract for each category a number of contour segments, namely, proto-segs $\{\kappa\}$, from the prototype shape examples, which are selected as the *means* after applying k-means algorithm on all the positive examples. Thus, the prototype examples are the representatives of the shape category, as shown in the left-hand side of Fig. 2(a). We then describe each proto-seg κ with three shape distances corresponding to the various shape deformations. In the three metric spaces, each parameterized proto-seg κ is further spanned into a number of subspaces, namely, ϵ-balls, as follows:

$$\Omega^{w}(\kappa) = \{c | D^{w}(c, \kappa) < \epsilon\} \tag{1}$$

where $w \in \{'p', 'a', 'g'\}$ indicates the type of shape distance metrics, i.e., procrustes distance, articulation distance, and geodesic distance, as shown in Fig. 2(a). Each ϵ-ball can be viewed as an equivalent class bounded by residual ϵ, in which all the contour segments $\{c\}$ are transformation invariant with respect to κ.

One ϵ-ball can be transformed to a binary feature (weak classifier) with the response of feature (output of the classifier) defined as

$$r(\mathbf{S}; \kappa, \epsilon) = \begin{cases} 1, & \mathcal{D}^{w}(c, \kappa) < \epsilon, \exists c \in \mathbf{S} \ \text{ s.t. } \theta_c \approx \theta_\kappa \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

where $\theta_c \approx \theta_\kappa$ indicates that two contour segments, c and κ, have similar positions related to the shapes.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

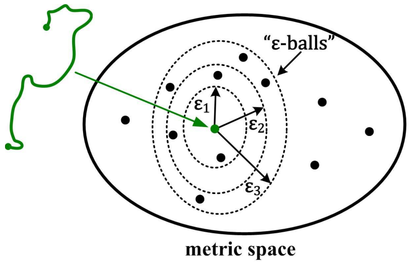


Fig. 4. In the step of feature evolution, all the proto-segs are mapped to the metric space. And we grow three ϵ-balls for each proto-seg. The three ϵ's are determined by the ball containing 0.1%, 0.3%, and 0.5% of total proto-segs.

Intuitively, (2) shows that a shape $\mathbf{S}$ can be predicted as positive if there exists a contour segment c in $\mathbf{S}$ such that, compared with the proto-seg κ, both c and κ have similar positions and are similar under the distance metric ω. Unlike the discriminative boundaries in many previous works [32], [38] that output $+1$ and -1, the proposed features have zero responses to a shape $\mathbf{S}$ not falling into them.

A. Feature Evolution

To describe ϵ-balls with the distance metrics, the residual ϵ needs to be further determined. We introduce an empirical procedure called *feature evolution* for this step. For each proto-seg κ in a certain metric space, i.e., $\omega \in \{'p', 'a', 'g'\}$, we generate three ϵ-balls, i.e., to compute ϵ_1, ϵ_2, and ϵ_3, which are determined as follows. First, we map all the proto-segs in the metric space, where each of them can be considered as a point, as the dots shown in Fig. 4. Second, for a proto-seg (green line), we grow the ϵ starting from an initially small value, and then the ϵ-ball may cover more neighbors when the ϵ increases. Three ϵ's are determined by the ball containing 0.1%, 0.3%, and 0.5% of total proto-segs. The above values of the percentages are small enough so that each ϵ-ball contains a very small number of neighbors to maintain its discriminativeness. Moreover, these values are chosen empirically and are fixed through all the experiments, including both the data sets with binary images and edge maps.

To achieve a compact feature representation, we prune those redundant ϵ-balls with high relevance, i.e., covering similar neighbors. We define the correlation of two arbitrary ϵ-balls as

$$\mathrm{corr}(\Omega(\kappa_i) \mid \Omega(\kappa_j)) = \frac{|\Omega(\kappa_i) \cap \Omega(\kappa_j)|}{|\Omega(\kappa_j)|}. \qquad (3)$$

Note that (3) is a nonsymmetric measure. For example, if $\Omega(\kappa_2) \subset \Omega(\kappa_1)$, then $\mathrm{corr}(\Omega(\kappa_1) \mid \Omega(\kappa_2)) = 1$ and $\mathrm{corr}(\Omega(\kappa_2) \mid \Omega(\kappa_1)) < 1$. We prune any ϵ-ball $\Omega(\kappa_i)$ with $\mathrm{corr}(\Omega(\kappa_i) \mid \Omega(\kappa_j)) > \gamma$, where γ is empirically set as 0.8 in all our experiments. Intuitively, our approach prefers the ϵ-balls with smaller sizes because they are more discriminative.

B. Model Pursuit via Information Projection

In this section, we introduce our learning algorithm that pursues the generative shape model with the ϵ-balls. For each shape category, given a training set $\{(\mathbf{S}_1, l_1), \ldots, (\mathbf{S}_N, l_N)\}$, where $l \in \{1, 0\}$ denotes the label of example: 1 indicates the positive example and 0 the reference example chosen from all the other categories. Recall that we use k-means on the positive examples to obtain the prototype shapes. To further group positive samples into different views, we adopt a hierarchical clustering method with the average linkage on the previous results of k-means. For most of the data sets used in this paper, grouping the samples into three to five views is enough. More implementation details are discussed in Section V. In the following, we mainly describe the learning procedure.

Let $f(\mathbf{S})$ denote the underlying probability distribution or target model for each shape category, from where the positive examples are sampled, and let $q(\mathbf{S})$ be an initial model or reference model, which is characterized by the reference examples. Our objective is to learn a series of models $p_k(\mathbf{S})$ that approach $f(\mathbf{S})$ from $q(\mathbf{S})$ by greedily choosing and matching features, i.e., the responses of ϵ-balls as introduced in (2). Intuitively, given a set of ϵ-balls as features $\{r_i\}$,[1] at each iteration of our algorithm, we first choose a feature r that best separates the positives and the reference examples, and then compute the weight of this feature by solving $E_{p_k}[r] = E_f[r]$, which means that the kth pursued model $p_k(\mathbf{S})$ should have the same empirical expectation on feature r with respect to the target model $f(\mathbf{S})$.

In addition, we introduce the information projection principle [10], [39], which ensures the convergence of our learning process.

Proposition 1: The pursued models $p_{k-1}(\mathbf{S})$ and $p_k(\mathbf{S})$ and the target model $f(\mathbf{S})$ satisfy the following equation:

$$\begin{aligned}\mathrm{KL}(f(\mathbf{S}) \| p_{k-1}(\mathbf{S})) - \mathrm{KL}(f(\mathbf{S}) \| p_k(\mathbf{S}))\\ = \mathrm{KL}(p_k(\mathbf{S}) \| p_{k-1}(\mathbf{S})) > 0 \qquad (4)\end{aligned}$$

where $\mathrm{KL}(\cdot \parallel \cdot)$ denotes the Kullback–Leibler divergence. Equation (4) shows that the procedure converges when the information gain, the right-hand side of (4), approaches zero. To this end, we devise an algorithm iterating between two steps, namely, MaxMin-KL. In the Max-KL step, we maximize $\mathrm{KL}(p_k(\mathbf{S}) \| p_{k-1}(\mathbf{S}))$ to select the most informative feature r that can best tell the difference between $E_f[r] = E_{p_k}[r]$ and $E_q[r] = E_{p_{k-1}}[r]$. Notice that at the kth iteration, the current shape model $f(\mathbf{S})$ is approximated by $f_k(\mathbf{S})$, and the reference model is by the latest model $p_{k-1}(\mathbf{S})$. In the Min-KL step, we minimize $\mathrm{KL}(p_k(\mathbf{S}) \| p_{k-1}(\mathbf{S}))$ to solve the feature's weight.

From the above discussions, our shape model can be expressed as the following Gibbs distribution form:

$$p_k(\mathbf{S}) = p_1(\mathbf{S}) \prod_{k=1}^{K} \frac{1}{Z_k} \exp\{\lambda_k r_k(\mathbf{S})\} \qquad (5)$$

where $p_1(\mathbf{S}) = q(\mathbf{S})$ and Z_k is a normalizing term that is estimated by $Z_k = E_q[\exp\{\lambda_k r_k(\mathbf{S})\}]$. λ_k is the feature's weight and thus is found by $E_f[r_k] = E_{p_k}[r_k]$.

Here, we discuss the MaxMin-KL iterations in more detail for solving r_k and λ_k in (5). Note that the number of iterations

[1] We simplify the notation of $r(\mathbf{S}; \kappa, \epsilon)$ as r.

can be estimated by the classification rate on a validation set in practice.

1) *Max-KL:* This step is to optimize the following problem:

$$r_k^* = \arg\max_{\{r_k\}} \mathrm{KL}(p_k(\mathbf{S}) \| p_{k-1}(\mathbf{S})) \tag{6}$$

$$= \arg\max_{\{r_k\}} \lambda_k E_f[r_k] - \log Z_k. \tag{7}$$

Following [10], (6) can be approximated by:

$$\arg\max_{\{r_k\}} \lambda_k E_f[r_k] - \log Z_k \cong \arg\max_{\{r_k\}} E_f[r_k] - E_q[r_k] \tag{8}$$

which encourages a feature in the target model but not in the reference model. We simply calculate the empirical expectations by the mean response values as

$$E_f[r_k] = \frac{1}{N_v^+} \sum_{i=1}^{N_v^+} r_k(\mathbf{S}_i) \tag{9}$$

$$E_q[r_k] = \frac{1}{N^-} \sum_{i=1}^{N^-} r_k(\mathbf{S}_i). \tag{10}$$

Here, N_v^+ represents the number of positive examples in the same view as the ϵ-ball r_k and N^- denotes the number of reference examples.

2) *Min-KL:* After the selection of a feature r_k, this step is to compute its corresponding Z_k and λ_k so that the constraint $E_f[r_k] = E_{p_k}[r_k]$ is satisfied. We conclude that

$$Z_k = e^{\lambda_k} E_q[r_k] + 1 - E_q[r_k] \tag{11}$$

$$\lambda_k = \log \frac{E_f[r_k](1 - E_q[r_k])}{(1 - E_f[r_k])E_q[r_k]}. \tag{12}$$

A proof is given in the Appendix. For the extreme case that $E_f[r_k] = E_q[r_k]$, which means feature r_k is not informative at all, then λ_k becomes zero. Otherwise, if $E_f[r_k] > E_q[r_k]$, then r_k has positive weight.

We summarize the sketch of learning shape models in Algorithm 1.

1) Extension: The proposed algorithm is simple and fast, because the values of $E_f[r_k]$ and $E_q[r_k]$ only need to be computed once for each feature r_k on the training examples. A possible way to improve the feature selection is that at each step, we update $E_f[r_i]$ and $E_q[r_i]$ for all the other features $\{r_{i \backslash k}\}$, except the selected one r_k, at each iteration by the correlation introduced in Section II-A as

$$E_f[r_i] = \frac{1}{N_v^+}(1 - \mathrm{corr}(r_k|r_i)) \sum_{j=1}^{N_v^+} r_i(\mathbf{S}_j) \tag{13}$$

$$E_q[r_i] = \frac{1}{N^-}(1 - \mathrm{corr}(r_k|r_i)) \sum_{j=1}^{N^-} r_i(\mathbf{S}_j). \tag{14}$$

This tends to select the features that are uncorrelated and discriminative. Section V demonstrates the effectiveness of such an extension.

```
Algorithm 1: Learning Compositional Shape Model
Input: A set of training examples,
    $\mathbf{T} = \{(\mathbf{S}_1, l_1), ..., (\mathbf{S}_N, l_N)\}$, where $l \in \{1, 0\}$
    denotes the positive and reference example
    respectively.
1. Prototype selection: apply k-means ($k = 15\sim30$)
on $\mathbf{T}$, the "prototype" shapes are selected as the means,
as shown in the beginning of section II.
2. Feature generation: extract a set of proto-segs, $\{\kappa\}$,
from the "prototype" shapes, and map them into three
distance matrices as in section III.
3. Feature evolution: specify the subspaces of each $\kappa$ by
determining the radiuses $\epsilon_1, \epsilon_2, \epsilon_3$, as in section II-A.
    3.1: Pose each subspace as a binary feature as Equ. (2).
    3.2: Prune the highly correlated features by Equ. (3).
Loop k=1 to K
    Max-KL: select $r_k$ by Equ. (8) and (9); update
    $E_f[r_i]$, $E_q[r_i]$ for the other features $r_i$ by Equ. (13).
    Min-KL: calculate $\lambda_k$, $Z_k$ by Equ. (11).
Output: A probability shape model defined in Equ. (5).
```

2) Discussion: The shape model can be synthesized from the distribution of (5) by the Hamiltonian Monte Carlo [40], where a key step is to compute the gradient of the energy function

$$\sum_{k=1}^{K} \lambda_k \partial r_k(c), \quad c \in \mathbf{S} \tag{15}$$

where c is the contour segment corresponding to the kth feature and is represented by the different distance metrics $\mathcal{D}^w(c, \kappa)$, which are extensively discussed in Section III. For computation simplicity, we assume that $r_k(c)$ is a classifier predicted soft score

$$r_k(c) = \max(0, \epsilon - \mathcal{D}^w(c, \kappa)). \tag{16}$$

Then, by the sampling process, we can obtain the value of c under the certain distance metric $\mathcal{D}^w$.

For model visualization, we search for the contour segment from all positive shapes that has a similar value with c in the corresponding feature space. In this way, we simply find the contour segments related with all the features of the model, and thus obtain the synthesized the shape. Some illustrative examples will be presented in the experiments. The detailed method of model synthesis goes beyond our scope, and we refer to [10] and [11] for the theoretical background.

III. Distance Metrics of Contour Segments

In this section, we introduce three distance metrics combined in the procedure of shape model pursuit, including procrustes distance, articulation distance, and geodesic distance.

A. Procrustes Distance Metric

The procrustes distance is defined as the sum of the squared distances over features of corresponding points on contours c and κ

$$\mathcal{D}^p(c, \kappa) = \|\overline{c} - \overline{\kappa}\|_2 \tag{17}$$

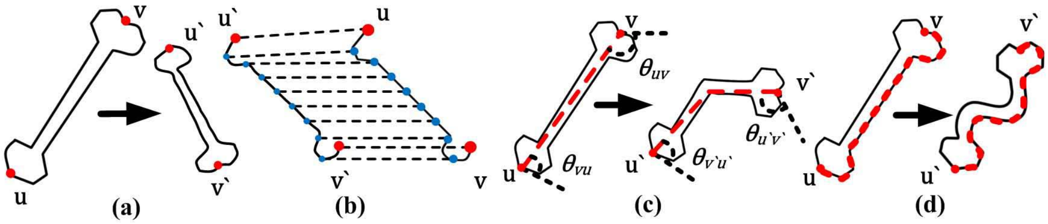


Fig. 5. Illustrations of different shape distance metrics for capturing deformations. (a) Rigid transformation between two bones. (b) Bijection correspondence between two contour segments using procrustes analysis. (c) and (d) Articulated transformation and distortion (twist) between two shapes, respectively.

where $\overline{c}$ and $\overline{\kappa}$ are the feature vectors of c and κ, obtained by concatenating the shape descriptors extracted from every key point on the contour. Therefore, the problem becomes how to search the point correspondences between two contours, which is not trivial because different contours have different lengths, rotations, and scales. We adopt the procrustes matching method [4], which first translates and scales both contours to the origin and unit lengths, and then estimate a rotation matrix that makes these two contours having the largest overlap, which is to minimize the following problem:

$$\min_{\alpha,\Gamma,\beta} \frac{1}{N} \sum_{n=1}^{N} \|\kappa_n - (\alpha \Gamma c_n + \beta)\|_2^2 \tag{18}$$

where α is a scaling parameter, $\beta \in \mathbb{R}^2$ is a 2-D translation vector, and $\Gamma \in \mathbb{R}^{2\times 2}$ is a 2-by-2 rotation matrix. Equation (18) has a simple closed form expression as follows:

$$\mathcal{D}^p(c,\kappa) = \frac{\left(\overline{\kappa} * \overline{\kappa} - \frac{\overline{\kappa} * \overline{c}\,\overline{c} * \overline{\kappa}}{\overline{c} * \overline{c}}\right)}{N} \tag{19}$$

in which we represent each point as a complex number, $(x_n, y_n) = x_n + iy_n$, and $\overline{\kappa}$ and $\overline{c} \in \mathbb{C}^N$ are two vectors of these complex numbers. One example of this metric is shown in Fig. 5(a), showing the rigid transformation between two bones, and Fig. 5(b) shows the bijection correspondences of two contour segments. After we retrieved the correspondences between the contours, we can extract features on the key points and compute (17). In the following, we consider the angle matrix [37] and shape context [41] as the features, which are both rotation, translation, and scaling invariant.

1) Angle Matrix: This contour descriptor calculates the angle of every two points on the contour with respect to a reference point. Specifically, we compute a matrix M with all the diagonal values equaled to zeros and every entry m_{ij}, $i \neq j$, defined by the angle between a line connecting the points (x_i, y_i) and (x_j, y_j) and a line from (x_j, y_j) to the reference point, which is chosen as the center of the bounding box of the contour.

2) Shape Context: The shape context histogram is also extracted from each point on the contour. In our implementation, we use two bins for spacial distance and six bins for orientation from 0 to 2π. Thus, the feature vector of each point is $2 \times 6 = 12$.

B. Articulation Distance Metric

In order to address the articulation problem shown in Fig. 5(c), we devise the articulation distance metric by combining three geometrical shape descriptors. This distance metric is defined similarly as (17), but employs different features that are articulation invariant. Here, $\overline{c}$ and $\overline{\kappa}$ denote two 6-D feature vectors computed using the following three shape descriptors on the corresponding contour segments.

1) Inner Distance Between the End Points: This descriptor computes the inner distance between the end points u, v of a contour segment, as shown in the red dashed line in Fig. 5(c). Following [8], we first build an undirected graph with the points of the contour segment as its nodes, and then apply the shortest route algorithm, e.g., Bellman–Ford, over this graph to compute the shortest length between two points.

2) Relative Angles: As shown in Fig. 5(c), the relative angles are θ_{uv} and θ_{vu} and are both stable under articulation, $\theta_{uv} \approx \theta_{u'v'}$ and $\theta_{vu} \approx \theta_{v'u'}$.

3) Articulated-Invariant Curve Signature: Let d^{in} and d^{eu} denote two matrices, whose elements are the inner distances and Euclidean distances between each pair of points on a contour segment, respectively. Given a set of points Φ of a contour segment and $d_{\Phi}^{\text{in}} \in \mathbb{R}^{|\Phi|\times|\Phi|}$, which is a matrix of inner distances based on Φ, this descriptor finds a set of transformed points Φ' so that $d_{\Phi'}^{\text{eu}}$ equals d_{Φ}^{in} in an element-wise manner. We adopt multidimensional scaling to find Φ' by minimizing the following equation:

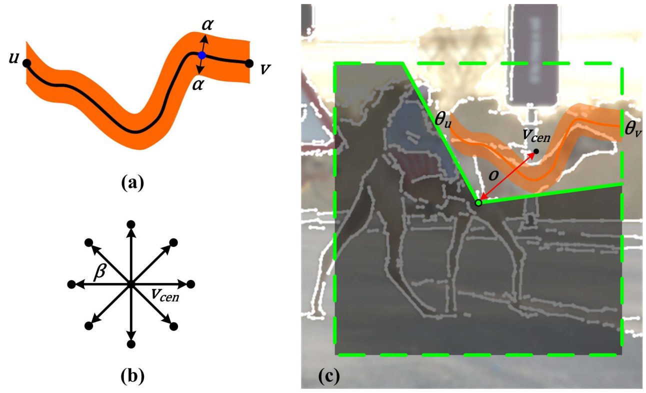
$$\Phi'^{*} = \arg\min_{\Phi'} \sum_{i}^{|\Phi|} \sum_{j}^{|\Phi|} \frac{\left(d_{\Phi}^{\text{in}}(i,j) - d_{\Phi'}^{\text{eu}}(i,j)\right)^2}{d_{\Phi}^{\text{in}}(i,j)^2} \tag{20}$$

which can be minimized using the scaling by maximizing a convex function (SAMCOF) algorithm [42]. More details can be found in [5].

Based on Φ', our articulated-invariant curve signature is defined to be a triple, which includes the distances between $\langle u', v' \rangle$, $\langle u', v'_{\text{cen}} \rangle$, and $\langle v'_{\text{cen}}, v' \rangle$, where $u', v' \in \Phi'$ and v'_{cen} is the center of Φ'.

C. Geodesic Distance Metric

Our motivation for this descriptor comes from [5] using geodesic distances for 3-D surface comparison. In 3-D space, the geodesic distance between any pair of points on a surface is defined as the length of the shortest path on the surface between them. In 2-D space, if two 2-D shapes to be matched have an identical view and a similar size, the counterpart of the geodesic distance in 2-D is the distance between a pair of points along the contour as shown in Fig. 5(d). Thus, $\mathcal{D}^g(c,\kappa)$ is defined as the Euclidean distance between the contour length of c and the length of the proto-seg κ.



(a)

(b)

(c)

Fig. 6. Illustration for shape detection by our model. (a) Proto-seg. We detect object contours from the edge map inside a ribbon of each proto-seg. (b) Idea of moving the ribbon around its eight neighbors. (c) Example of evaluating proto-seg in an image.

IV. Model-Based Shape Detection

The shape model in (5) can be transformed into a shape detector as

$$H(\mathbf{S}) = \log \frac{p_K(\mathbf{S})}{q(\mathbf{S})} = \sum_{k=1}^{K} (\lambda_k r_k(\mathbf{S}) - \log Z_k). \qquad (21)$$

We then detect shapes by introducing a threshold γ ($\gamma = 0$ in our implementation) over $H(\mathbf{S})$. In particular, if $H(\mathbf{S}) \geq \gamma$, the shape $\mathbf{S}$ is predicted to be positive; otherwise, $H(\mathbf{S}) \leq \gamma$.

To detect shapes at different scales, we adopt the coarse-to-fine sliding window approach [38]. Given a window, the shape model is used as a deformable template to match shape in an image, which is to evaluate (2) for each chosen r_k. Recall that we encode the spatial configuration, i.e., $\Theta_u, \Theta_v, v_{\text{cen}}$[2] for each proto-seg, as described in Section III. Thus, we only need to evaluate the proto-seg in a specific location, rather than in every location of the window. Furthermore, since the object boundary in cluttered edge map is usually broken and surrounded by noise, we can match the proto-seg within a small region.

There are three steps to match each proto-seg.

1) We detect curves in the image inside a ribbon of κ, illustrated as a region in Fig. 6(a).
2) The ribbon is moved around eight positions to detect curves, as shown in Fig. 6(b).
3) The value of r_k is determined by the curve that has a minimum distance with the proto-seg.

V. Experiments

We apply our approach in the following tasks: 1) shape classification using binary images; 2) shape detection from cluttered edge maps; and 3) shape-based image categorization. In addition, we justify the generation ability of our model by the synthesis experiments.

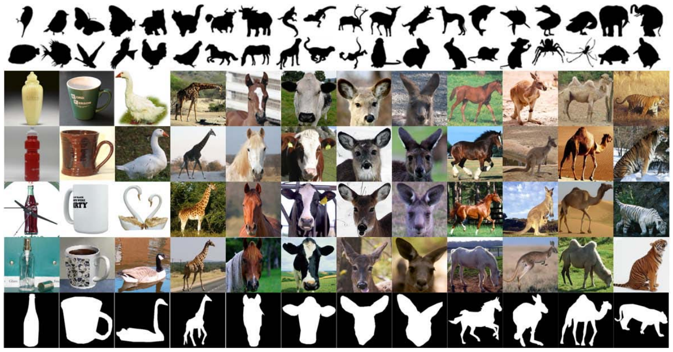


Fig. 7. Few binary images of shapes from the data set with 20 animal classes proposed in [14] are shown in the first two rows. Last five rows are several data of the data set with 40 image categories chosen from LHI database [16]. The last row shows some label maps.

A. Experiment I: Shape Classification

We first evaluate our approach on a challenging data set proposed in [14], which includes 20 categories, 100 binary images of each category, with totally 2000 animal shapes. A few examples are shown in the first two rows of Fig. 7, in which the large intraclass variances of this data set are demonstrated. This data set contains more examples on each shape category than the well-known MPEG-7 shape database [43] that has 20 images for each of 70 categories, and thus it is more suitable for testing learning based method.

Preprocessing: We uniformly sample 200 points for each shape, translate it to zero (the origin of coordinate), and then rescale it to 256 × 256 with its aspect ratio preserved. One of the challenges of this data set is that rotation is induced by hand on each example to increase its difficulty. However, this rotation can be reduced by estimating the first principle component of each shape using PCA, which is similar to compute the eigenvectors of the covariance matrix of the data. The only difference is that our space is in two dimensions, where each point represents a 2-D landmark on the shape contour, rather than an example in the high-dimensional space. Thus, we can align all the shapes using their first principle components.

In our method, 50 shapes for each category are selected for training and the other 50 are for testing. We first apply k-means to pick up 10 prototype shapes as introduced in Section III. Then, we extract 80 proto-segs with their lengths in the range of $[80, 120]$ points. Thus, there are 80 × 3 metrics × 3 epsilons × 10 prototype shapes = 7200 features. After feature pruning in Section II-A, we finally obtain about 6000 features. In addition, for each shape category, we randomly select 10 shapes from the training set of the other categories as the reference examples. At the beginning of the training step, we cluster the positive training examples into four views. Moreover, the MPEG-7 database [43] is adopted as the validation set to tune the number of features K in our model. The average value of K is 300 on 20 categories in this experiment.

We compare with the class segment [1], inner distance+shape context (IDSC) [8], curvature scale space (CSS) [23], Fourier descriptor [17], contour segment + skeleton (CS + SP) [14], bag of contour (BC) [44], and

[2] v_{cen} of the shape is estimated by the center of the window.

TABLE I
AVERAGED CLASSIFICATION ACCURACY ON THE ANIMAL DATA SET [14]

Methods	Class Segment[1]	IDSC[8]	CSS[23]	Fourier[17]	CS+SP[14]	L&G[45]	BC[44]	Ours (no ext.)	Ours
Accuracy	70.2%	73.9%	67.5%	62.6%	79.2%	80.37%	84.3%	83.0%	**88.2**%

TABLE II
CLASSIFICATION RATES OF DIFFERENT COMBINATIONS OF THE PARAMETERS. THE COMBINATION IS DENOTED IN THE FORM AS (THE NUMBER OF PROTOTYPE SHAPES)−(THE NUMBER OF VIEWS)−(THE NUMBER OF FEATURES K IN THE MODEL)

	10-5-300	10-4-300	10-3-300	10-2-300	10-1-300
Accuracy	87.0%±1.2%	**87.2**%±**1.0**%	85.6%±1.3%	84.9%±0.8%	83.4%±0.4%
	5-4-300	8-4-300	12-4-300	15-4-300	
Accuracy	60.2%±1.5%	77.7%±0.6%	**87.2**%±**1.3**%	85.5%±0.6%	
	10-4-100	10-4-400	10-4-600	10-4-800	10-4-1000
Accuracy	76.9%±1.7%	**82.4**±**2.2**%	77.0%±0.8%	71.1%±0.7%	70.6%±0.6%

local and global features [45]. The implementations of IDSC,[3] CSS,[4] CS + SP,[5] and BC[6] are publicly available. We implement the class segment method and Fourier descriptor with MATLAB. All these methods adopt the same preprocessing step if necessary as discussed above. Table I shows the average accuracies. The accuracies of our approach without and with extension (Section II) are 83.0% and 88.2%, respectively, both of which outperform all other baseline methods. In the rest of the experiments, we report only the results of our method with extension.

In Table II, we also evaluate our approach regarding different combinations of the parameters, including the number of prototype shapes, the number of views, and the number of features K in the model. We repeated the experiment 10 times and reported the mean and variance of the accuracies. There are several interesting results. First, the overall classification rates decreased when reducing the number of views, because the selected proto-segs cannot capture the intraclass variances. For example, a proto-seg appears in all positives (with view variance) may lead to the largest information gain, if there is only one view. This proto-seg may not be discriminative among views. However, the accuracy becomes saturate when the number of views is larger than five, because the number of training samples for each view is limited. Second, decreasing the number of prototype shapes hurts the results, e.g., accuracy of 5-4-300 drops 27% compared with 10-4-300. However, increasing it does not help much since the size of the positive training examples becomes small. Therefore, one may need to balance the ratio between prototype shapes and positive training samples. Finally, adding more features to the shape model tends to overfit the training data.

B. Experiment II: Shape Detection

We evaluate our method on two data sets: 1) the ETHZ image data set [9] containing 255 images with their probability edge maps and 2) the UIUC-People data set [15] containing 593 images, which are very challenging due to large shape variations caused by different views and human poses.

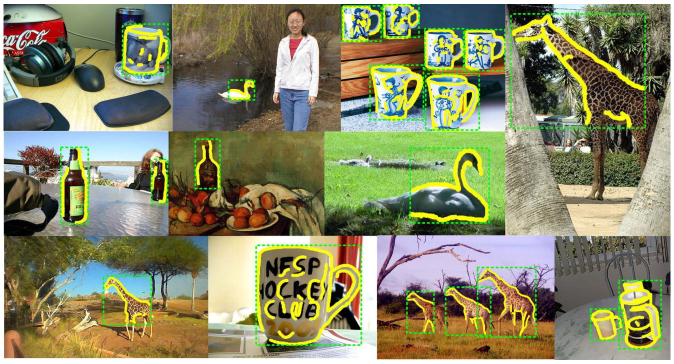


Fig. 8. Some selected results on the ETHZ data set [9] with the detected bounding boxes and contours plotted in green and yellow, respectively.

Preprocessing: On the ETHZ data set, we select four classes (i.e., *Bottles*, *Giraffes*, *Mugs*, and *Swans*). For each class, we randomly partition the images into half for training and half for testing. In the training stage, the binary images (label maps) are manually annotated.

As the number of samples for each category is small, we select 30 shapes from the LHI database [16] as prototype shapes. Then, about 100 proto-segs with their lengths in the range of [30, 120] are extracted on each prototype shape. Note that we use a larger range in ETHZ than the range in MPEG-7 shape database. Since the images in ETHZ data set include clutterred backgrounds, the detected contours are probably broken. Large range enables the pursuit of discriminative short contours. There are more than 10^4 features after the feature pruning. The maximum number of the selected features K is set to be 500 in the model pursuit. On the UIUC-People data set, we use 346 images for training and 247 for testing, following the same splitting standard in [15] and [46]. We randomly select 50 prototype instances from the training set, from which we extract the proto-segs. The other setting is the same as on the ETHZ data set. In these experiments, our method takes only about 5–10 min to learn the shape models, on a PC with 4-G RAM, Intel Core i5 CPU 3.3 GHz.

[3] IDSC: http://www.dabi.temple.edu/~hbling/code_data.htm
[4] CSS: http://www.mathworks.com/matlabcentral/
[5] CS + SP: https://sites.google.com/site/xiangbai/
[6] BC: https://bitbucket.org/xinggangw/bcf

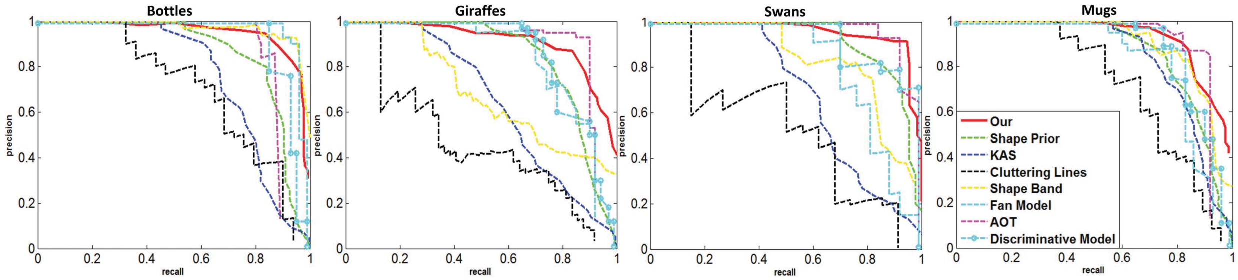


Fig. 9. Comparison of PR curves for four classes on ETHZ [9].

TABLE III
PRECISIONS ARE COMPARED WITH [9], [29], [30], [32], [35], [36], [47], AND [48] AT THE SAME RECALL RATES

	Bottles	Giraffes	Swans	Mugs
Our precision/recall (%)	85/93 **50/97**	88/70 **46/97**	**90/94** 58/97	**85/83 60/97**
AOT [35]	85/93 –/–	**95/70** –/–	78/94 **64/97**	**84/83** –/–
Discriminative Model [36]	78/93 13/97	**92/70** 12/97	**80/94 70/97**	74/83 13/97
Fan Model [47]	**94/93** 44/97	82/70 –/–	32/94 16/97	38/83 –/–
Shape Prior [48]	40/93 8/97	89/70 16/97	60/94 38/97	70/83 20/97
Shape Band [29]	**95/93 55/97**	56/70 **34/97**	44/94 38/97	83/83 **38/97**
Cluttering Lines [30]	41/93 –/–	38/70 –/–	20/94 –/–	40/83 10/97
KAS [9]	11/93 9/97	39/70 10/97	14/94 16/97	64/83 13/97
KAS08 [32]	33/93 –/–	44/70 –/–	23/94 –/–	41/83 –/–

TABLE IV
DETECTION ACCURACIES ON THE UIUC-PEOPLE DATA SET [15]

Method	Accuracy
Our approach	0.674
And-Or Graph [46]	**0.680**
Hierarchical Poselets [49]	0.668
Relational Models [15]	0.663
Latent-SVM [50]	0.586

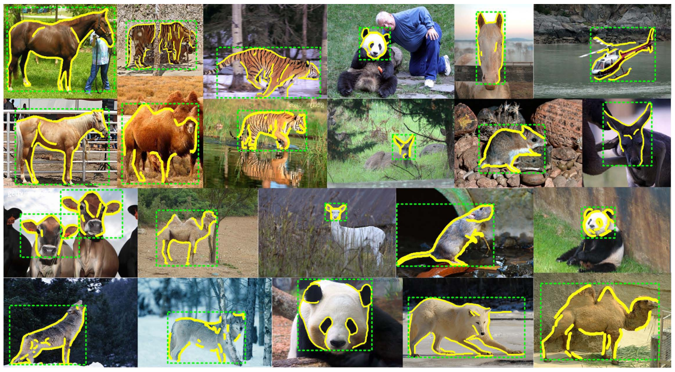


Fig. 10. We show some selected results on the LHI database [16] and demonstrate that the shape detectors are robust to various deformations, background clutter, and occlusion.

In the testing stage, we obtain edge maps of the testing images by Canny detector, and then scan detection windows with different scales as in [38]. Furthermore, the shape matching algorithm described in Section IV is applied in each sliding window. The radius α in Fig. 6 is set as 15, and the radius β in Fig. 6 is 10. It takes about 15 s to process an image on a PC as above.

Fig. 8 shows some representative results on this data set with the detected bounding boxes shown in green and localized curves in yellow by our system. We compare our result with shape prior [47], shape band [29], clustering lines [30], KAS [9], fan model [48], AND–OR tree (AOT) model [35], and the discriminative shape model [36], using the precision–recall (PR) curves. All the above methods are learning-based methods except shape band. Furthermore, shape prior, clustering lines, fan model, and our method are the generative learning methods, while AOT and discriminative shape model adopt discriminative learning. The results are summarized in Fig. 9. We also report precisions of the above methods at the same recall rates in Table III, where the first and the second best performing methods are highlighted. Table III shows that the discriminative methods work better than generative methods in some complex scenarios, such as *Giraffes* and *Swans*. Our model, however, achieves comparatively good results in all the four categories by incorporating multiple shape metrics. Our approach outperformed all existing works when the recall rate is high, e.g., 97%, where the accuracies of most of the methods drop significantly.

On the UIUC-People data set, we also demonstrate very promising results compared with other state-of-the-art methods, and Table IV reports the quantitative detection accuracies generated by our method and the competing approaches [46], [49], [50]. It is worth mentioning that [49] relies on complicated appearance features and a manually annotated model structure and [46] requires a good initialization for model training.

C. Experiment III: Shape-Based Image Categorization

We further evaluate our method on a image data set with 40 categories selected from the LHI database [16], which

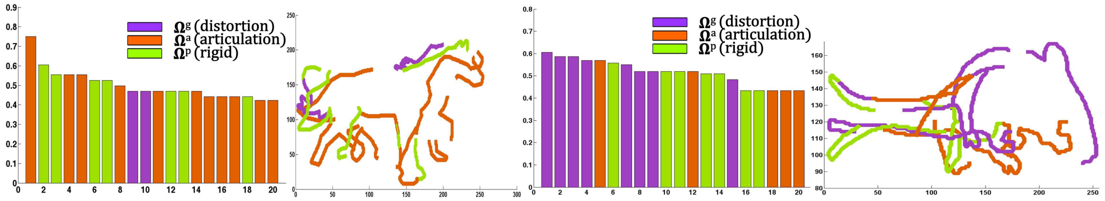


Fig. 11. Top 20 most informative features of *Horse* (left) and *Mouse* (right) are visualized. Different colors indicate three distance metrics. From these results, we conclude that horses are more likely to perform articulation and mice are usually distorted, which matches our intuition very well. Moreover, we find that articulation mostly occurs on four limbs, while distortion happens more often on the back and tail of animals. The shape models consisting of ϵ-balls can be viewed as the implicit deformable templates that include different local shape variances.

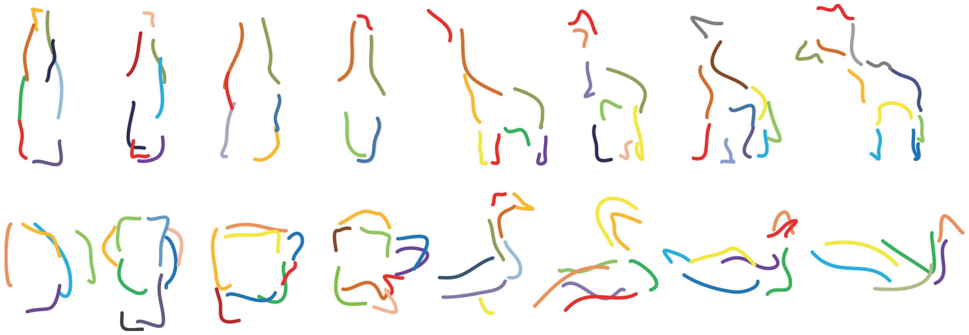


Fig. 12. Several synthesized examples from the learned shape models.

is publicly available.[7] Each category has 90 images with 3600 images in total. Some shapes and label maps are shown in Fig. 7. For each image, we obtain its edge map by the contour detection algorithm [51]. Our task is to detect and classify shapes from the edge maps. Due to the heavy occlusion, shading, and surrounding clutter, this task is more complicate. For example, the animal faces in Fig. 7 are hardly distinguished.

Preprocessing: For each shape category, we separate the images into half for training and half for testing. Then, 15 positive examples are chosen as the prototype shapes. We finally group the positives into five views. The additional settings are similar to Experiment II.

We compare our result with generative model [52], active skeleton [53], fan model [48], AOT model [35], and BC [44]. The experiment is conducted using the publicly available code of the above methods. Our approach obtained an overall classification rate as 88.5%, which outperforms the 77.2% of active skeleton [53], 81.4% of generative model [52], 82.1% of BC [44], 83.4% of fan model [48], and 86.1% of AOT [35]. Fig. 10 shows several selected results on this data set.

D. Experiment IV: Model Visualization

In this experiment, we visualize the learned shape models and showed that which types of features are effective with regard to different categories. We use two categories, *Horse* and *Mouse*, from the data in Experiment III. As shown in Fig. 11, we sample top 20 informative features using the strategy discussed in Section II. Features with different distance metrics are denoted by different colors, green for procrustes metric Ω^p, orange for articulation metric Ω^a, and purple for geodesic metric Ω^g. The results show that horses are more likely to perform articulation and mice are usually distorted, which matches our intuitive observation very well. Moreover, we also present some illustrative examples that are synthesized from the learned shape models in Fig. 12. These results well demonstrate one of the advantages of our model, i.e., the synthesis results make our model very intuitive and understandable, compared with the traditional discriminative classifiers.

VI. Conclusion

This paper incorporated three types of shape distance metrics to learn compositional shape models. Our model utilized parameterized contour segments to form the implicit deformable templates, which account for different shape variances. Extensive experiments demonstrated that our method significantly improved the shape classification and detection results on several public data sets.

The proposed algorithm is very general. Different shape descriptors and distance metrics can be adaptively incorporated into our framework. Also, the model pursuit algorithm can evaluate the informativeness of shape distance metrics for each shape category.

Our future work will focus on three aspects.

1) We intend to develop more expressive structure for contour-based shape modeling, e.g., AND–OR graph model.
2) Our shape matching algorithm in Section IV that evaluates one proto-seg at a time can be made parallelized. In this case, our algorithm can be generalized to a large-scale problem.

[7] http://www.imageparsing.com/FreeDataOutline.html

3) We will explore more applications, such as recognizing sketches drawn by the artists, shape retrieval, and shape-based image retrieval. To this end, we may study more effective shape distance metrics into our framework.

APPENDIX

Proof of (11) and (12): After the selection of the most informative feature r_k in the kth Max-KL step, we find the corresponding λ_k and Z_k by satisfying the constraint $E_f[r_k] = E_{p_k}[r_k]$, where $E_f[r_k]$ is approximated by the positive training examples. The proof follows the notations in Algorithm 1 as below:

$$\begin{aligned}
E_{p_k}[r_k] &= \sum_{\forall \mathbf{S} \in \mathbf{T}} p_k(\mathbf{S}) r_k && (22)\\
&= \sum_{\forall \mathbf{S} \in \mathbf{T}} p_{k-1}(\mathbf{S}) \frac{1}{Z_k} \exp\{\lambda_k r_k\} r_k && (23)\\
&= \frac{1}{Z_k} \sum_{\{\mathbf{S} | D^w(\mathbf{S}) \geq \epsilon\}} p_{k-1}(\mathbf{S}) \exp\{\lambda_k r_k\} r_k && (24)\\
&\quad + \frac{1}{Z_k} \sum_{\{\mathbf{S} | D^w(\mathbf{S}) < \epsilon\}} p_{k-1}(\mathbf{S}) \exp\{\lambda_k r_k\} r_k\\
&= \frac{1}{Z_k} \exp\{\lambda_k\} E_{p_{k-1}}[r_k]. && (25)
\end{aligned}$$

For any shape $\mathbf{S}$, if $D^w(\mathbf{S}) \geq \epsilon^8$, then $r_k = 0$; otherwise, $r_k = 1$. Therefore, (24) can be written as (25). Moreover, following (5), $Z_k = E_q[\exp\{\lambda_k r_k\}]$, which has the following form if we apply the same trick:

$$\begin{aligned}
Z_k &= E_q[\exp\{\lambda_k r_k\}] && (26)\\
&= \sum_{\forall \mathbf{S} \in \mathbf{T}} p_{k-1}(\mathbf{S}) \exp\{\lambda_k r_k\} && (27)\\
&= \sum_{\{\mathbf{S} | D^w(\mathbf{S}) \geq \epsilon\}} p_{k-1}(\mathbf{S}) \exp\{\lambda_k r_k\} && (28)\\
&\quad + \sum_{\{\mathbf{S} | D^w(\mathbf{S}) < \epsilon\}} p_{k-1}(\mathbf{S}) \exp\{\lambda_k r_k\}\\
&= \sum_{\{\mathbf{S} | D^w(\mathbf{S}) \geq \epsilon\}} p_{k-1}(\mathbf{S}) && (29)\\
&\quad + \sum_{\{\mathbf{S} | D^w(\mathbf{S}) < \epsilon\}} p_{k-1}(\mathbf{S}) \exp\{\lambda_k\}\\
&= \exp\{\lambda_k\} E_{p_{k-1}}[r_k] + 1 - E_{p_{k-1}}[r_k]. && (30)
\end{aligned}$$

Note that we approximate $E_q[r_k]$ using $E_{p_{k-1}}[r_k]$ at the kth step as in Section II, that is why (26) can be expanded as (27).

Let $E_f[r_k] = E_{p_k}[r_k]$ and combine (25) with (30)

$$\lambda_k = \log \frac{E_f[r_k](1 - E_q[r_k])}{(1 - E_f[r_k]) E_q[r_k]} \qquad (31)$$

$$Z_k = \exp\{\lambda_k\} E_q[r_k] + 1 - E_q[r_k]. \qquad (32)$$

[8] We simplify the notation $D^w(c, \kappa)$ for the selected feature r_k as $D^w(\mathbf{S})$.

REFERENCES

[1] K. B. Sun and B. J. Super, "Classification of contour shapes using class segment sets," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, San Diego, CA, USA, Jun. 2005, pp. 727–733.

[2] P. F. Felzenszwalb and J. D. Schwartz, "Hierarchical matching of deformable shapes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Minneapolis, MN, USA, Jun. 2007, pp. 1–8.

[3] L. Lin, X. Wang, W. Yang, and J.-H. Lai, "Discriminatively trained And-Or graph models for object shape detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 5, pp. 959–972, May 2015.

[4] G. McNeill and S. Vijayakumar, "Hierarchical Procrustes matching for shape retrieval," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 1. Jun. 2006, pp. 885–894.

[5] A. Elad and R. Kimmel, "On bending invariant signatures for surfaces," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 10, pp. 1285–1295, Oct. 2003.

[6] I. L. Dryden and K. V. Mardia, *Statistical Shape Analysis*. New York, NY, USA: Wiley, 1998.

[7] C. Goodall, "Procrustes methods in the statistical analysis of shape," *J. Roy. Statist. Soc.*, vol. 53, no. 2, pp. 285–339, 1991.

[8] H. Ling and D. W. Jacobs, "Shape classification using the inner-distance," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 2, pp. 286–299, Feb. 2007.

[9] V. Ferrari, F. Jurie, and C. Schmid, "Accurate object detection with deformable shape models learnt from images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Minneapolis, MN, USA, Jun. 2007, pp. 1–8.

[10] S. C. Zhu, Y. N. Wu, and D. Mumford, "Minimax entropy principle and its applications to texture modeling," *Neural Comput.*, vol. 9, no. 8, pp. 1627–1660, 1997.

[11] J. Xie, W. Hu, S.-C. Zhu, and Y. N. Wu, "Learning sparse FRAME models for natural image patterns," *Int. J. Comput. Vis.*, Oct. 2014, doi: 10.1007/s11263-014-0757-x.

[12] L. Lin, P. Luo, X. Chen, and K. Zeng, "Representing and recognizing objects with massive local image patches," *Pattern Recognit.*, vol. 45, no. 1, pp. 231–240, 2012.

[13] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. New York, NY, USA: Wiley, 1991.

[14] X. Bai, W. Liu, and Z. Tu, "Integrating contour and skeleton for shape classification," in *Proc. IEEE 12th ICCV Workshop*, Sep./Oct. 2009, pp. 360–367.

[15] D. Tran and D. Forsyth, "Improved human parsing with a full relational model," in *Proc. 11th Eur. Conf. Comput. Vis.*, Heraklion, Greece, Sep. 2010, pp. 227–240.

[16] B. Yao, X. Yang, and S.-C. Zhu, "Introduction to a large-scale general purpose ground truth database: Methodology, annotation tool and benchmarks," in *Energy Minimization Methods in Computer Vision and Pattern Recognition*. Berlin, Germany: Springer-Verlag, 2012, pp. 169–183.

[17] D. Zhang and G. Lu, "A comparative study on shape retrieval using Fourier descriptors with different shape signatures," *J. Vis. Commun. Image Represent.*, vol. 14, no. 1, pp. 41–60, 2003.

[18] T. B. Sebastian, P. N. Klein, and B. B. Kimia, "Recognition of shapes by editing their shock graphs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 5, pp. 550–571, May 2004.

[19] G. Tzimiropoulos, N. Mitianoudis, and T. Stathaki, "A unifying approach to moment-based shape orientation and symmetry classification," *IEEE Trans. Image Process.*, vol. 18, no. 1, pp. 125–139, Jan. 2009.

[20] E. Klassen, A. Srivastava, W. Mio, and S. H. Joshi, "Analysis of planar shapes using geodesic paths on shape spaces," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 3, pp. 372–383, Mar. 2004.

[21] D. D. Hoffman and W. A. Richards, "Parts of recognition," *Cognition*, vol. 18, nos. 1–3, pp. 65–96, 1984.

[22] P. Luo, L. Lin, and H. Chao, "Learning shape detector by quantizing curve segments with multiple distance metrics," in *Proc. 11th Eur. Conf. Comput. Vis. (ECCV)*, Heraklion, Greece, Sep. 2010, pp. 342–355.

[23] F. Mokhtarian and S. Abbasi, "Shape similarity retrieval under affine transforms," *Pattern Recognit.*, vol. 35, no. 1, pp. 31–41, 2002.

[24] L. Lin, X. Liu, and S.-C. Zhu, "Layered graph matching with composite cluster sampling," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 8, pp. 1426–1442, Aug. 2010.

[25] A. Kolesnikov and P. Fränti, "Polygonal approximation of closed contours," in *Proc. 13th Scandin. Conf. Image Anal. (SCIA)*, vol. 2749. 2003, pp. 778–785.

[26] F. L. Bookstein, "Principal warps: Thin-plate splines and the decomposition of deformations," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 11, no. 6, pp. 567–585, Jun. 1989.

[27] J. Duchon, "Splines minimizing rotation-invariant semi-norms in Sobolev spaces," in *Constructive Theory of Functions of Several Variables*, vol. 571. Berlin, Germany: Springer-Verlag, 1977, pp. 85–100.

[28] C. Gu, J. J. Lim, P. Arbelaez, and J. Malik, "Recognition using regions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Miami, FL, USA, Jun. 2009, pp. 1030–1037.

[29] X. Bai, Q. Li, L. J. Latecki, W. Liu, and Z. Tu, "Shape band: A deformable object detection approach," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 1335–1342.

[30] B. Ommer and J. Malik, "Multi-scale object detection by clustering lines," in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, Kyoto, Japan, Sep./Oct. 2009, pp. 484–491.

[31] Q. Zhu, L. Wang, Y. Wu, and J. Shi, "Contour context selection for object detection: A set-to-set contour matching approach," in *Proc. 10th Eur. Conf. Comput. Vis.*, Marseille, France, Oct. 2008, pp. 774–787.

[32] V. Ferrari, L. Fevrier, F. Jurie, and C. Schmid, "Groups of adjacent contour segments for object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 1, pp. 36–51, Jan. 2008.

[33] P. Srinivasan, Q. Zhu, and J. Shi, "Many-to-one contour matching for describing and discriminating object shape," in *Proc. 23rd IEEE Conf. Comput. Vis. Pattern Recognit.*, San Francisco, CA, USA, Jun. 2010, pp. 1673–1680.

[34] J. Shotton, A. Blake, and R. Cipolla, "Multiscale categorical object recognition using contour fragments," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 7, pp. 1270–1281, Jul. 2008.

[35] L. Lin, X. Wang, W. Yang, and J. Lai, "Learning contour-fragment-based shape model with And-Or tree representation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Providence, RI, USA, Jun. 2012, pp. 135–142.

[36] P. Yarlagadda and B. Ommer, "From meaningful contours to discriminative object shape," in *Proc. 12th Eur. Conf. Comput. Vis.*, Florence, Italy, Oct. 2012, pp. 766–779.

[37] P. Kontschieder, H. Riemenschneider, M. Donoser, and H. Bischof, "Discriminative learning of contour fragments for object detection," in *Proc. Brit. Mach. Vis. Conf.*, Dundee, U.K., Aug./Sep. 2011, pp. 1–12.

[38] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2005, vol. 1, no. 1, pp. 886–893.

[39] S. Della Pietra, V. Della Pietra, and J. Lafferty, "Inducing features of random fields," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 4, pp. 380–393, Apr. 1997.

[40] R. M. Neal, "MCMC using Hamiltonian dynamics," in *Handbook of Markov Chain Monte Carlo*. Boca Raton, FL, USA: CRC Press, 2011.

[41] S. Belongie, J. Malik, and J. Puzicha, "Shape matching and object recognition using shape contexts," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 4, pp. 509–522, Apr. 2002.

[42] I. Borg and P. J. F. Groenen, *Modern Multidimensional Scaling: Theory and Applications*. New York, NY, USA: Springer-Verlag, 1997.

[43] L. J. Latecki and R. Lakämper, "Shape similarity measure based on correspondence of visual parts," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 10, pp. 1185–1190, Oct. 2000.

[44] X. Wang, B. Feng, X. Bai, W. Liu, and L. J. Latecki, "Bag of contour fragments for robust shape classification," *Pattern Recognit.*, vol. 47, no. 6, pp. 2116–2125, 2014.

[45] K.-L. Lim and H. K. Galoogahi, "Shape classification using local and global features," in *Proc. 4th Pacific-Rim Symp. Image Video Technol. (PSIVT)*, Nov. 2010, pp. 115–120.

[46] X. Wang and L. Lin, "Dynamical And-Or graph learning for object shape modeling and detection," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2012, pp. 242–250.

[47] T. Jiang, F. Jurie, and C. Schmid, "Learning shape prior models for object matching," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Miami, FL, USA, Jun. 2009, pp. 848–855.

[48] X. Wang, X. Bai, T. Ma, W. Liu, and L. J. Latecki, "Fan shape model for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Providence, RI, USA, Jun. 2012, pp. 151–158.

[49] Y. Wang, D. Tran, and Z. Liao, "Learning hierarchical poselets for human parsing," in *Proc. 24th IEEE Conf. Comput. Vis. Pattern Recognit.*, Colorado Springs, CO, USA, Jun. 2011, pp. 1705–1712.

[50] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1627–1645, Sep. 2010.

[51] P. Arbeláez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 5, pp. 898–916, May 2011.

[52] L. Lin, S. Peng, J. Porway, S.-C. Zhu, and Y. Wang, "An empirical study of object category recognition: Sequential testing with generalized samples," in *Proc. IEEE 11th Int. Conf. Comput. Vis.*, vol. 1. Oct. 2007, pp. 1–8.

[53] X. Bai, X. Wang, L. J. Latecki, W. Liu, and Z. Tu, "Active skeleton for non-rigid object detection," in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, Kyoto, Japan, Sep./Oct. 2009, pp. 575–582.

Ping Luo received the B.S. and master's degrees from Sun Yat-Sen University, Guangzhou, China, in 2008 and 2010, respectively, and the Ph.D. degree in information engineering from The Chinese University of Hong Kong, Hong Kong, in 2014.

His current research interests include computer vision and machine learning, with a focus on deep learning, face analysis, and large-scale object recognition and detection.

Liang Lin received the B.S. and Ph.D. degrees from the Beijing Institute of Technology, Beijing, China, in 1999 and 2008, respectively.

He was a joint Ph.D. Student with the Department of Statistics, University of California at Los Angeles (UCLA), Los Angeles, CA, USA, from 2006 to 2007. He was a Post-Doctoral Research Fellow with the Center for Vision, Cognition, Learning, and Art, UCLA. He is currently a Professor with the School of Advanced Computing, Sun Yat-Sen University, Guangzhou, China. His current research interests include new models, algorithms, and systems for intelligent processing and understanding of visual data, such as images and videos. He has authored over 70 papers in top tier academic journals and conferences. He was supported by several promotive programs or funds for his works, such as Program for New Century Excellent Talents through the Ministry of Education, China, in 2012, and Guangdong NSFs for Distinguished Young Scholars in 2013.

Prof. Lin received the Best Paper Runners-Up Award from ACM NonPhotorealistic Animation and Rendering in 2010, the Google Faculty Award in 2012, and the Best Student Paper Award from the IEEE International Conference on Multimedia and Expo in 2014. His Ph.D. dissertation achieved the China National Excellent Ph.D. Thesis Award Nomination in 2010. He has served as an Associate Editor for *Neurocomputing* and *The Visual Computer*.

Xiaobai Liu received the Ph.D. degree from the Huazhong University of Science and Technology, Wuhan, China.

He is currently a Post-Doctoral Research Fellow with the Center for Vision, Cognition, Learning and Art, University of California at Los Angeles, Los Angeles, CA, USA. He has authored over 30 peer-reviewed articles in top-tier conferences and leading journals. His current research interests include scene parsing with a variety of topics, e.g., joint inference for recognition and reconstruction, and commonsense reasoning.

Dr. Liu received a number of awards for his academic contribution, including the 2013 Outstanding Thesis Award from the China Computer Federation.