

A Multi-scale Generative Model for Animate Shapes and Parts

Aleksandr Dubinskiy and Song Chun Zhu
dubinskiy1@yahoo.com, sczhu@stat.ucla.edu

Abstract

This paper presents a multi-scale generative model for representing animate shapes and extracting meaningful parts of objects. The model assumes that animate shapes (2D simple closed curves) are formed by a linear superposition of a number of shape bases. These shape bases resemble the multi-scale Gabor bases in image pyramid representation, are well localized in both spatial and frequency domains, and form an over-complete dictionary. This model is simpler than the popular B-spline representation since it does not engage a domain partition. Thus it eliminates the interference between adjacent B-spline bases, and becomes a true linear additive model. We pursue the bases by reconstructing the shape in a coarse-to-fine procedure through curve evolution. These shape bases are further organized in a tree-structure where the bases in each subtree sum up to an intuitive part of the object. To build probabilistic model for a class of objects, we propose a Markov random field model at each level of the tree representation to account for the spatial relationship between bases. Thus the final model integrates a Markov tree (generative) model over scales and a Markov random field over space. We adopt EM-type algorithm for learning the meaningful parts for a shape class, and show some results on shape synthesis.

1. Introduction

In this paper, animate shapes refer to 2D simple closed curves which are the boundaries of deformable objects, such as animals and plants. These shapes have been studied extensively for image segmentation, object recognition, object tracking, and medical image analysis. In the literature, there are two types of representation, one is region-based, epitomized by medial axis representation, and the other is contour-based, exemplified by polynomial, PCA, and B-splines.

Region-based representation. The *medial axis* representation is very powerful and has achieved impressive success in recognition and morphing[16, 12, 6]. The representation proposed in this paper is complementary to the medial axis, and will be especially suitable

for representing shapes with bulky bodies and unclear skeletal structures. It is a step toward integrating the region based and contour based representations.

Contour-based representations. Let $\Gamma(s) = (x(s), y(s))$ $s \in [0, 1]$ be the 2D shape. A contour-based method represents $\Gamma(s)$ by a superposition of bases. For example, in the explicit polynomial formulation one uses monomials of the form $1, s, s^2, \dots, s^n$ as bases. Thus a curve is represented in a generative way as

$$\Gamma(s) = \sum_{k=0}^K \begin{bmatrix} x_k \\ y_k \end{bmatrix} (1-s)^{n-k} s^k + \mathbf{n}(s). \quad (1)$$

where $\mathbf{n}(s)$ is the noise. Generally, one can also use Fourier bases a similar expression[14].

The PCA bases are very popular for representing a category of shapes[3, 5]

$$\Gamma(s) = \Gamma_o(s) + \sum_{k=0}^K a_k \mathbf{b}_k(s) + \mathbf{n}(s). \quad (2)$$

where Γ_o is the average shape and each $b_k(s)$ is a curve that captures the direction of deformation in the object's biology. The PCA basis is promising but is meaningful only within a particular category if there are well established landmarks[1, 5] on the contour and the deformations do not involve structural changes.

The polynomial, Fourier, and PCA representations are well criticized in literature for having global bases and lacking locality. One representation that makes a step in the direction of improved locality is *B-splines*.

In a B-spline representation, the domain $[0, 1]$ is broken up into K intervals by a partition

$$\pi = (s_0, s_1, \dots, s_K), s_0 < s_1 < \dots < s_K.$$

Each B-spline base $B_k^d(s; \pi)$ spans d -intervals in the partition with d being the degree of the spline. Thus the representation is

$$\Gamma(s) = \sum_{k=1}^K \begin{bmatrix} x_k \\ y_k \end{bmatrix} B_k^d(s; \pi) + \mathbf{n}. \quad (3)$$

Here (x_k, y_k) are *control points*. The B-spline representation appears to be linear additive, but its bases

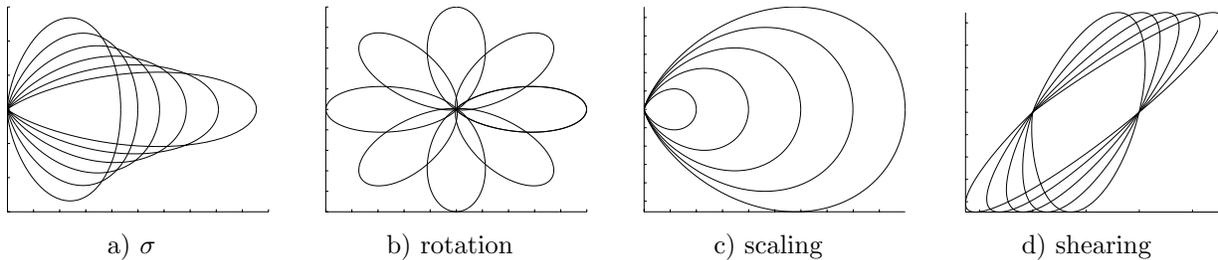


Figure 1: Each shape base is a lobe-shaped curve. It displays variety with different parameter and transforms.

depend on the underlying partition π . Adding or deleting a base will change the partition and thus affect the forms of several adjacent bases, so the model's linearity is lost. Furthermore, when we add more control points, the scopes of existing bases are broken to small intervals, and thus lack the concept of multi-scale representation.

Our objective: Motivated by the above observations, we present a multi-scale linear additive representation.

Firstly, we construct an over-complete dictionary of bases. Motivated by the success of image pyramid representation[8, 13] and the philosophy of over-complete basis[10], we construct bases which are sine and cosine waves modulated by Gaussians at various variances. Each base is a pair of such 1D Gabor sine and cosine components, for example, Fig.1 shows some instances of the bases at different scales subject to transforms. As the Gaussian variance $\sigma \rightarrow \infty$, we obtain a special base – the ellipse.

Secondly, we use curve evolution equations[9] to derive curves at a coarse-to-fine sequence. At the coarse level, all simple curves evolve to an ellipse. The curve reconstruction starts with the ellipse, and proceeds to the next level by adding new bases. The representation is linear additive, so the new bases do not disturb the existing ones. Thus we have bases at multiple scales, which are visualized as a “shape script” (see Figs.2 and 3), which looks similar to music script.

Thirdly, we organize the bases in the shape script in a tree structure, so that the bases in each subtree sum up to an intuitive part of the object (see Fig.3). This achieves some decomposition of the shape.

Fourthly, to build probabilistic model for a class of objects, we propose a Markov random field model at each level of the tree representation to account for the spatial relationship between bases. Thus the final model integrates a Markov tree (generative) model over scales and a Markov random field over space.

We adopt a match-pursuit[7, 8] type algorithm for selecting bases, and an EM-type algorithm for learning

the meaningful parts for a shape class, and demonstrate the power of our method using shape synthesis.

2. A linear curve representation

2.1. An over-complete shape basis

We sought a dictionary of localized and intuitive basis functions that would be general across a wide variety of shapes. Inspired by the success of Gabor and wavelets in signal reconstruction[8, 13], we choose the following formulation,

$$\psi(s; \mu, \sigma) \propto \exp\left(-\frac{(s - \mu)^2}{2\sigma^2}\right) \begin{bmatrix} \cos\left(\frac{2\pi}{\sigma}(s - \mu)\right) \\ \sin\left(\frac{2\pi}{\sigma}(s - \mu)\right) \end{bmatrix} \quad (4)$$

Here $\mu \in [0, 1]$ indicates the location of the basis function relative to the domain of the observed curve, and σ is the range of values s at which the function ψ is non-zero. In practice, the function is truncated to zero outside $[\mu - 1.5\sigma, \mu + 1.5\sigma]$. Fig.1.a shows the shape of the basis function ψ at different σ values. The basis functions are subject to affine transformations by a 2×2 matrix of *basis coefficients*

$$A_k = \begin{bmatrix} a_k & b_k \\ c_k & d_k \end{bmatrix}$$

The variables for describing a base are denoted by $\mathbf{b}_k = (A_k, \mu_k, \sigma_k)$ and are termed *basis elements*. We call the shape specified by \mathbf{b}_k its *shapelet* and denote it by

$$\gamma(s; \mathbf{b}_k) = A_k \psi(s; \mu_k, \sigma_k) \quad (5)$$

Figure 1 (b,c,d) demonstrates shapelets obtained from the basis functions ψ by the affine transformations of rotation, scaling, and shearing respectively, as indicated by the basis coefficient A_k . By collecting all the shapelets at various μ, σ, A and discretizing them at multiple levels, we obtain an over-complete dictionary

$$\Delta = \{\gamma(s; \mathbf{b}) : \forall \mathbf{b}; a \gamma_0, a > 0\}$$

A special shapelet γ_0 is defined as an ellipse.

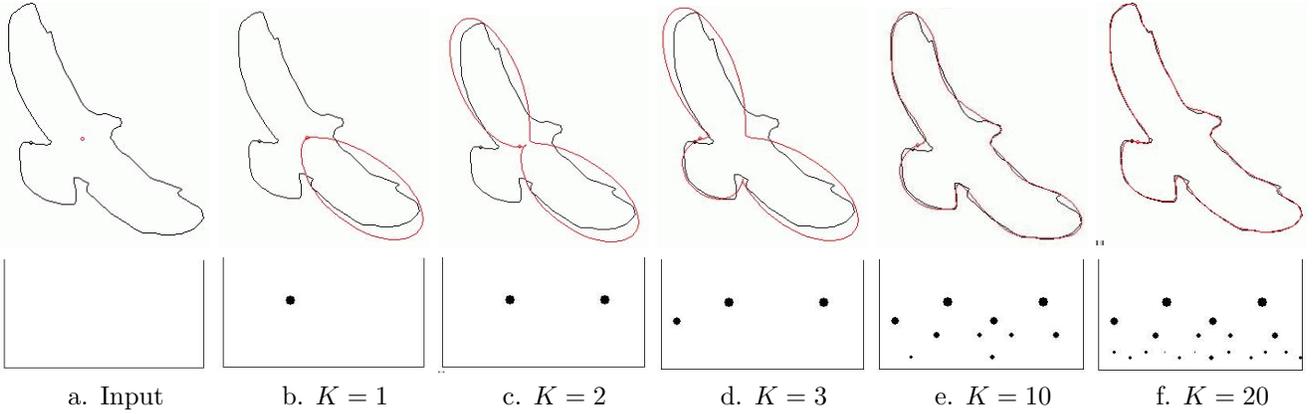


Figure 2: Pursuit of shape bases for an eagle contour. The second row shows the “shape script” in the (μ, σ) domain. The horizontal axis is $\mu \in [0, 1]$ and the vertical axis is the σ . Large dots mean big A_K^2 .

2.2. Linearly additive contour

Shapelets are the building blocks for shape contours, and they form closed curves by linear addition:

$$\Gamma(s) = \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} + \sum_{k=1}^K \begin{bmatrix} a_k & b_k \\ c_k & d_k \end{bmatrix} \psi(s; \mu_k, \sigma_k) + \mathbf{n}(s) \quad (6)$$

Here (x_0, y_0) is the mass center of the contour and \mathbf{n} is residue. A comparison to the formulation of the B-spline (eqn. 3) reveals that in the new formulation, shapelets no longer interfere with each other. Thus we do not have undesirable non-local perturbations when adding a base element. For example, Fig.2 shows the reconstruction of an eagle shape by adding the shapelets one by one. This transfers the continuous curve to a discrete representation $\mathbf{B} = (K, \mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_K)$, shown by the dots in second row of Fig.2. We call \mathbf{B} the “shape script” by analogy to music scripts, where each shapelet is represented by a dots in the (μ, σ) domain. The horizontal axis is $\mu \in [0, 1]$ and the vertical axis is the σ . Large dots correspond to big coefficient matrix $A_k^2 = a_k^2 + b_k^2 + c_k^2 + d_k^2$.

2.3. Pursuit of shapelets

Unlike in the case of Fourier or PCA, computing the shape script \mathbf{B} is a non-trivial task, since Δ is over-complete and there will be multiple sets of bases that reconstruct the curve with equal precision. Obviously, we would prefer a small number of bases. We adopt the matching pursuit procedure studied by Mallat-Zhang [7] for wavelet coding.

Recall a shapelet’s basis function has two orthogonal components $\psi(s) = (\psi_x(s), \psi_y(s))'$, $s \in [0, 1]$, we

normalize them to unit length,

$$\int_0^1 \psi_x^2(s) ds = \int_0^1 \psi_y^2(s) ds = 1, \quad \int_0^1 \psi_x(s) \cdot \psi_y(s) ds = 0.$$

Suppose $\Gamma^{\text{obs}}(s)$ is an input curve, $\Gamma(s; \mathbf{B})$ its current approximation with K shapelets in \mathbf{B} , and

$$\Gamma^{\text{res}}(s; \mathbf{B}) = (x^{\text{res}}(s; \mathbf{B}), y^{\text{res}}(s; \mathbf{B})) = \Gamma^{\text{obs}}(s) - \Gamma(s; \mathbf{B})$$

being the residue. Let $\mathbf{B}_+ = \mathbf{B} \cup \{\mathbf{b}_{K+1}\}$ with an new shapelet. We want to choose \mathbf{b}_{K+1} for the largest reduction of the approximation error,

$$\mathbf{b}_{K+1}^* = \arg \min_{\mathbf{b}_{K+1} \in \Delta} \|\Gamma^{\text{obs}}(s) - \Gamma(s; \mathbf{B}_+)\|_2.$$

The L_2 -norm is computed as

$$\|\Gamma_1(s) - \Gamma_2(s)\|_2 = \int_0^1 (x_1(s) - x_2(s))^2 + (y_1(s) - y_2(s))^2 ds.$$

Since the components of any basis function $\psi(s; \mu_{K+1}, \sigma_{K+1})$ are orthogonal, it is not hard to show that the optimal transform (coefficient) matrix $A_{K+1} = (a_{K+1}, b_{K+1}; c_{K+1}, d_{K+1})$ can be computed deterministically as the inner products between the residual curve and the basis function,

$$\begin{aligned} a_{K+1} &= \int_0^1 x^{\text{res}}(s; \mathbf{B}) \psi_x(s; \mu_{K+1}, \sigma_{K+1}) ds \\ b_{K+1} &= \int_0^1 x^{\text{res}}(s; \mathbf{B}) \psi_y(s; \mu_{K+1}, \sigma_{K+1}) ds \\ c_{K+1} &= \int_0^1 y^{\text{res}}(s; \mathbf{B}) \psi_x(s; \mu_{K+1}, \sigma_{K+1}) ds \\ d_{K+1} &= \int_0^1 y^{\text{res}}(s; \mathbf{B}) \psi_y(s; \mu_{K+1}, \sigma_{K+1}) ds \end{aligned}$$

Note that this extends the coefficient in image coding to 2 bands. The amount of error reduction is shown to be the size of the coefficient matrix,

$$\begin{aligned} E(\mathbf{b}_{K+1}) &= \|\Gamma^{\text{res}}(s; \mathbf{B})\|_2 - \|\Gamma^{\text{res}}(s; \mathbf{B}_+)\|_2 \\ &= a_{K+1}^2 + b_{K+1}^2 + c_{K+1}^2 + d_{K+1}^2. \end{aligned}$$

Thus we compute A_{K+1} and choose \mathbf{b}_{K+1}^* to be the one which has largest error reduction, i.e. A_{K+1}^2 , until we reach a certain reconstruction error or a predefined number of basis elements.

We demonstrate the matching pursuit process on the example of an eagle in Fig.2. The input shape (far left) is reconstructed well with $K > 10$ bases/shapelets. Each shapelet is represented by a dot in the shape script below. The size of each dot is proportional to the "size" of A_k , which is $a_k^2 + b_k^2 + c_k^2 + d_k^2$.

Note that this is not a globally optimal solution, so we can improve the reconstruction without adding more basis elements. We do so by adjusting the basis elements (A_k, μ_k, σ_k) using the Gibbs sampler to further reduce reconstruction error.

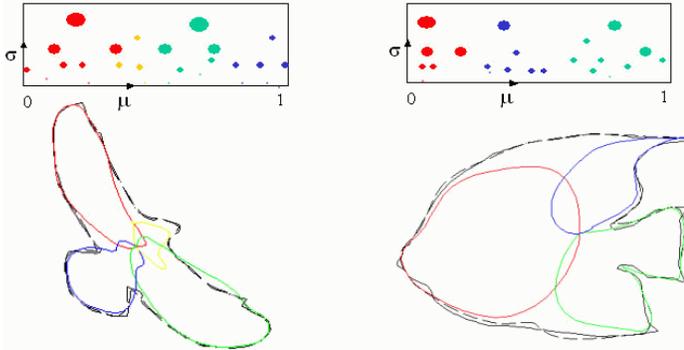


Figure 3: Subtrees and individual parts (please view color on screen)

2.4 Grouping shapelets into parts

It is not surprising that any over-complete dictionary Δ can reconstruct the shape well. A good dictionary should achieve reconstruction using a small number of bases and more importantly, the reconstruction should be decomposable into parts which are meaningful.

Fig.3 shows two examples of decomposability and parts. For the shape script of the eagle obtained above, we group the basis elements of \mathbf{B} into four groups $\mathbf{B}^1, \mathbf{B}^2, \mathbf{B}^3, \mathbf{B}^4$.

The bases in each subgroup sum up to a shape part

$$P^i(s) = \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} + \sum_{\mathbf{b}_j \in \mathbf{B}^i} \begin{bmatrix} a_j & b_j \\ c_j & d_j \end{bmatrix} \psi(s; \mu_j, \sigma_j), \quad i = 1..4.$$

The four parts correspond to the two wings, head, and tail respectively and are displayed in four different colors. Similarly the fish is decomposed into 3 parts.

This example shows that the linear representation is suitable for some class of shapes. In the next section we improve the shapelet pursuit procedure with a multi-scale scheme, and then show how to obtain parts and build probabilistic models on the shape scripts to account for shape deformations.

3 Multi-scale and parts

3.1. Curve evolution

Parts derived in section 2 are only one of many possible interpretations. In this section, we introduce a multi-scale reconstruction for a curve and its parts in a coarse-to-fine fashion. This is motivated by biological observations that animate objects, and thus their contours, "grow" from inside out, forming the main body first then the coarse features and finer features later.

This growth effect can be reproduced by curve evolution. Starting with the observed shape $\Gamma^{\text{obs}}(s)$, we run the following curvature flow equation[9, 6] to form a sequence of curves $C(s, t)$ over time t ,

$$\frac{dC(s, t)}{dt} = [\alpha + \beta\kappa(s)]n(s), \quad C(s, 0) = \Gamma^{\text{obs}}(s). \quad (7)$$

$n(s) = (y, -x)$ is the curve normal of $C(s, t)$ pointing inwards, and $\kappa(s)$ its curvature. $[x] = x$ if $x > 0$ and $[x] = 0$ if $x \leq 0$. Thus the curve always shrinks inwards. This defines a curve's scale-space [9].

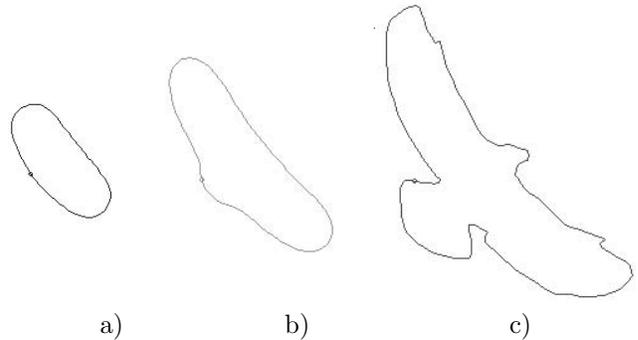


Figure 4: Contour of eagle at three scales

3.2. Coarse-to-fine reconstruction

In our experiments, we take three levels of the scale representation from coarse to fine

$$C(s, t_0), C(s, t_1), C(s, t_2); \quad t_0 > t_1 > t_2 = 0.$$

$C(s, t_0)$ is close to the ellipse represented by the special base γ_o , and $C(s, t_2) = \Gamma^{\text{obs}}(s)$. The middle and the innermost snapshots are taken according to the following intuition: Major parts, such as arms, wings, tails, often correspond to protrusions in the curve, which can be mathematically formalized as regions of convexity between regions of concavity. Thus we take the middle snapshot at the point where the number of concave regions is equal to the number of parts we are seeking.¹ The main body should have no protrusions, or 0 concave regions. This intuition works for a variety of animate image categories, regardless of the proportion of the main body to the parts. Fig.4 shows an example of the eagle at three time steps. Fig. 5 is a coarse-to-fine reconstruction of a cross. Fig. 5.a-c are respectively $C(s, t_0), C(s, t_1), C(s, t_2)$ at three sequential levels reconstructed by the shape scripts below. Now we obtain a tree-structured organization of the shapelets. Fig. 5.d shows a subtree which corresponds to one of the four parts. Note that the four subtrees share a central ellipse which serves as a joint.

To summarize, the shape script \mathbf{B} is organized in a tree structure.

(1). We can divide the tree into horizontal levels (or generations) $\mathbf{B} = (\mathbf{B}_0 = \gamma_o, \mathbf{B}_1, \mathbf{B}_2)$ from coarse-to-fine. This is similar to the Laplacian pyramid in image coding. Each level adds details to the shape, shown by red, green and blue respectively in Fig. 5.a-c.

(2). We can divide the tree vertically into subtrees, each corresponds to a part of shape.

Figure 7 shows nine examples of the vertical partitioning. Each subtree is plotted by a different color and corresponds to a curve which resembles the object parts, such as fins of fish, wings of butterfly and eagle, legs and trunk of the elephant.

4. Learning parts dictionary and deformable models

So far we have presented a multi-scale representation of animate curves and the decomposition into parts and bases. In this section, we explore two more ambitious goals:

(1). Learning generic probabilistic models on \mathbf{B} and thus on Γ for the purpose of recognition, segmentation, and tracking. The model should account for deformations and, more importantly, structural variations due to gesture and view point changes.

(2). Learning a generic dictionary of parts, built on top of the shapelet dictionary, for a broad range of

¹There is an inherent ambiguity in the number of parts, and at this early stage of research, we prespecify the number of parts for a particular image category.

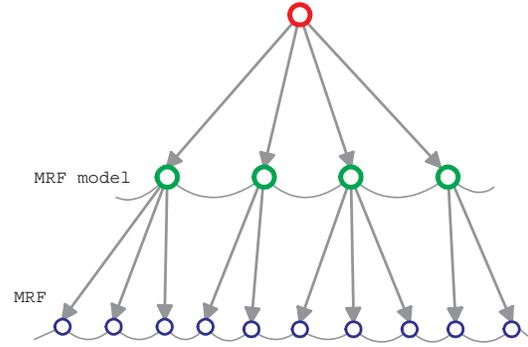


Figure 6: A probabilistic model of shape: a Markov tree model for multi-scale shape generation is integrated with Markov random field models at each horizontal level to account for spatial arrangement.

animate shapes.

4.1 Problem formulation

The problem is formulated as statistical learning. Suppose we are given a set of observed shapes $\{\Gamma_1^{\text{obs}}(s), \dots, \Gamma_M^{\text{obs}}(s)\}$, which are from the same object category for learning a deformable model, or are pooled from a range of animate objects for learning a generic dictionary of parts.

Now we have a two-stage generative model, a curve $\Gamma(s)$ is generated by a number of shape bases (or shapelets) \mathbf{B} selected from a dictionary Δ , and the shape bases are further generated by a tree \mathbf{T} of parts selected from a part dictionary Φ ,

$$\mathbf{T} \xrightarrow{\Phi} \mathbf{B} \xrightarrow{\Delta} \Gamma(s).$$

The dictionary Φ includes a number of production rules, like grammar in Language model, each rule is associated with a probability,

$$\Phi = \{r_i : \mathbf{b}_i \rightarrow \mathbf{b}_{i1} \mathbf{b}_{i2} \dots \mathbf{b}_{in_i}, p(r_i; \theta_i); \forall i\}.$$

$p(r_i; \theta_i) = (\mathbf{b}_{i1}, \mathbf{b}_{i2}, \dots, \mathbf{b}_{in_i} | \mathbf{b}_i; \theta_i)$ is a conditional probability defined on the attributes of each base $\mathbf{b} = (\mu, \sigma, A)$, specified by parameters θ_i . Thus a tree consist of a number of rules $\mathbf{T} = (r_1, r_2, \dots, r_n)$. This is basically a Markov tree process[4], and the probability of \mathbf{T} is

$$p(\mathbf{T}; \Theta) = \prod_{j=1}^n p(r_j; \theta_j).$$

$\Theta = (\theta_1, \theta_2, \dots, \theta_n)$. The general likelihood for a curve Γ is then

$$p(\Gamma; \Theta, \Phi) = \int p(\Gamma | \mathbf{B}; \Delta) p(\mathbf{B} | \mathbf{T}; \Phi) p(\mathbf{T}; \Theta) d\mathbf{B} d\mathbf{T}.$$

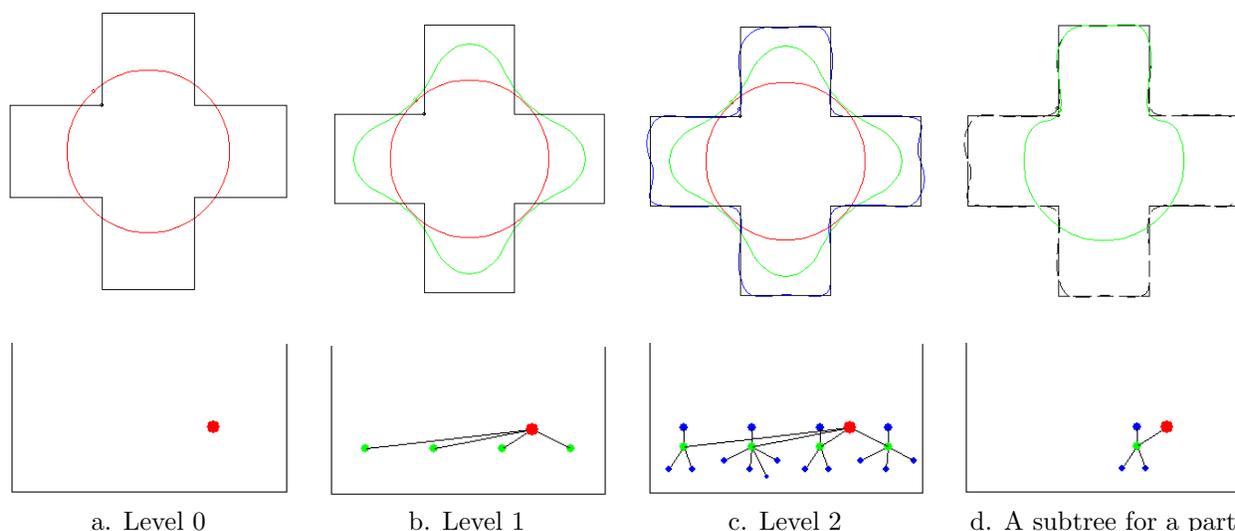


Figure 5: Coarse-to-fine Reconstruction (please view in color on screen)

In this formulation, \mathbf{B} , \mathbf{T} are hidden variables, like the parsing tree in language, associated with the specific shape Γ , while Φ, Θ are the parameters that govern the ensemble of shapes. The two objectives are then formulated as MLE-learning,

$$(\Phi, \Theta)^* = \arg \max \sum_{i=1}^M p(\Gamma_i^{\text{obs}}(s); \Theta, \Phi).$$

Remarks: (1). In the above formulation, we fixed the dictionary Δ , in general we can learn Δ from input shapes, this is similar to the learning of Gabor like image bases[10]. (2). The deformable model for each object is defined on the tree representation $p(\mathbf{T}; \Theta)p(\mathbf{B}|\mathbf{T}; \Phi)$ which can account for structural changes. (3). In a more general form, we should introduce a Markov random field model, or a simple Markov chain model for each horizontal level of the tree, as Fig.6 illustrates. The MRF or Markov chain model will account for local consistence between adjacent bases in the same level.

This formulation bear resemblance to texton learning[15] from images. Intuitively, the parts (sub-trees) correspond to the textons as the building blocks.

4.2 Learning and inference

The inference follows from an stochastic gradient algorithm which is a generalized version of the EM-algorithm. It proceeds in two iterative steps, as it is in the texton learning[15].

1. For each shape Γ_i^{obs} , infer the hidden tree structure from the posterior probability by Markov chain Monte Carlo sampling

$$(\mathbf{B}_i, \mathbf{T}_i) \sim p(\Gamma_i^{\text{obs}}|\mathbf{B}_i; \Delta)p(\mathbf{B}_i|\mathbf{T}_i; \Phi)p(\mathbf{T}_i; \Theta),$$

for $i = 1, 2, \dots, M$. In practice, there are many combinations of $(\mathbf{B}_i, \mathbf{T}_i)$ that can reconstruct Γ_i^{obs} well. Some supervised parsing will be very helpful for deriving meaningful parts, especially when the number of observed samples are not enough to favor some intuitive choice. We find that the coarse-to-fine reconstruction helps to introduce some order and structure in the reconstruction.

2. Given the samples above, we approximate the integration by importance sampling techniques, and then fit the parameters in Φ, Θ . In practice, this learning procedure can be expedited by a clustering step in the space of subtrees. For example, for a grammar rule r_i and conditional probability

$$p(r; \theta) = (\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_{k-1} | \mathbf{b}; \theta_i)$$

We form a polygon with $k+1$ vertices corresponding to bases

$$k\text{-gon} = (\mathbf{b}, \mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_{k-1}).$$

Collecting all the sub-trees like those in Fig.7, and each grammar rule in the subtrees, contribute a k -gon (i.e. a point) in the space of subtrees. By data clustering, such as K-means in the space of polygons, we got a

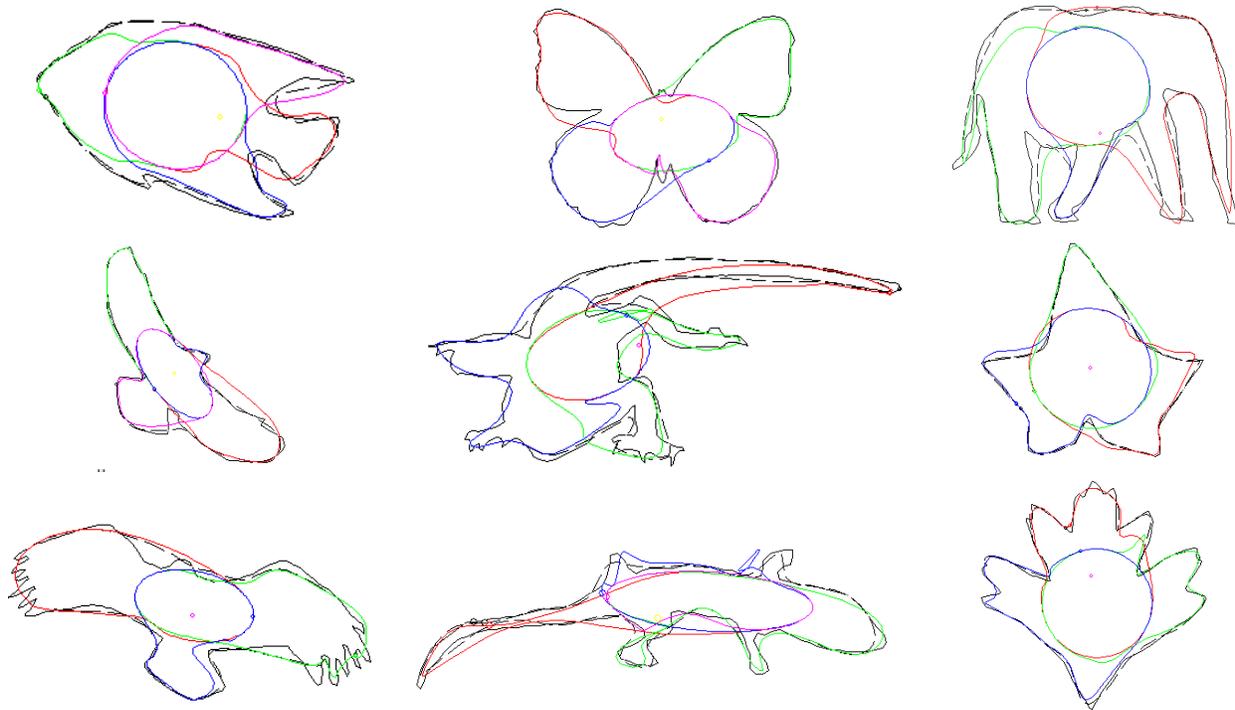


Figure 7: Subtrees and parts (in different colors) of animate shapes

number of clusters each corresponds to a part and is modeled by a probability.

Due to space limitation, we cannot unfold the details of the algorithm. Instead, we show two results of learning and sampling shape from the deformable model.

Figures 8.a and c show three examples of the observed leaves from two different categories of leaf, the observed leaves are parsed, and the parts are shown in colored curves. In Figures 8. b and d, we show three examples sampled from the Markov tree models. Clearly, such models can be used for segmentation, tracking, and shape morphing.

5. Future Work

The proposed representation is simple and combines the advantage of the linear additive representation, such as Fourier, polynomial and PCA, and the locality of B-splines. It is also multi-scale and decomposable to parts. This representation is especially suitable for blobby objects, and it has shortcomings in representing elongated objects. In future work, we should study hybrid models which combine the contour based and region based representation. That is the dictionary should include region-based shape elements, such as

the deformed primitive in the FORMS framework[16].

References

- [1] F. L. Bookstein, "Size and shape spaces for landmark data in two dimensions", *Statistical Science*, vol.1, No.2, 181-242, 1986.
- [2] H. Blum, "Biological shape and visual science", *J. of Theoretical Biology*, 33:205-287, 1973.
- [3] T. F. Cootes, D. Cooper, C.J. Taylor and J. Graham. "Active Shape Models - Their Training and Application," *Computer Vision and Image Understanding*, vol.61, no. 1, Jan. pages 38-59,1995.
- [4] M. Crouse, R. Nowak, and R. Baraniuk, "Wavelet-based statistical signal processing using hidden Markov models", *IEEE. Trans on. Signal Proc.*, 46, , 886-902, 1998.
- [5] I. L. Dryden and K. V. Mardia, *Statistical Shape Analysis*, Wiley Publisher, 1998.
- [6] B. Kimia, A.R. Tannenbaum, and S.W. Zucker, "Shapes, shocks, and deformations I: the com-

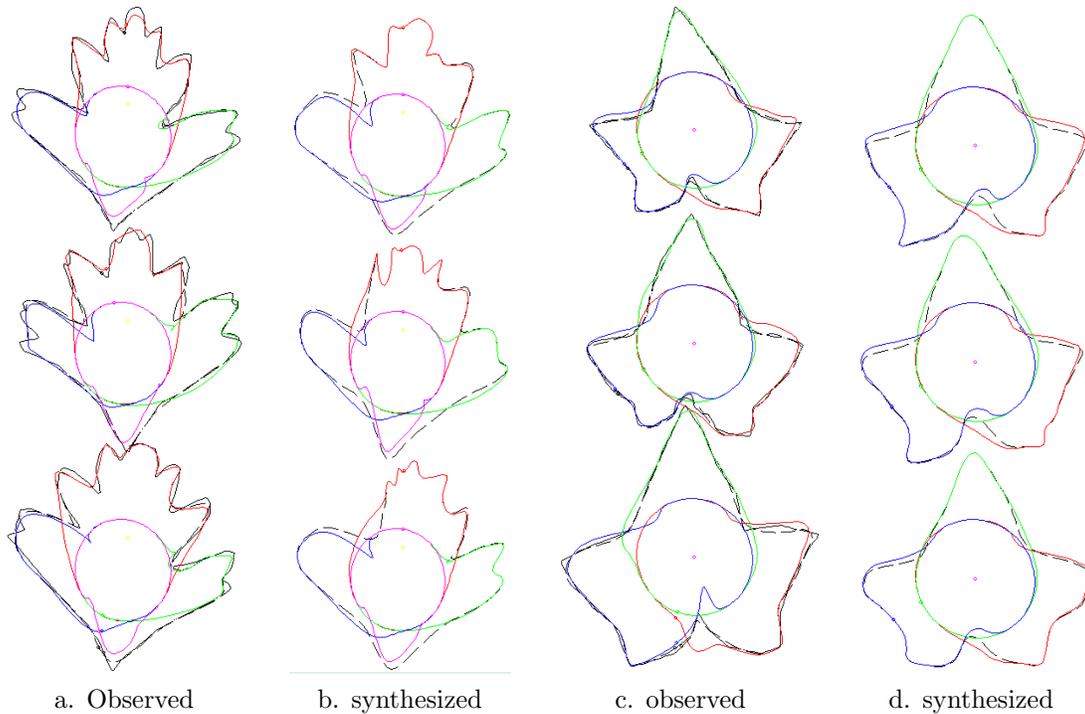


Figure 8: Two examples of shape synthesis by sampling from the probability models.

ponents of 2D shape and the reaction-diffusion space”, *IJCV*, August, 1995.

- [7] S.G. Mallat and Z. Zhang, “Matching Pursuits With Time-frequency Dictionaries,” *IEEE Trans. on Signal Processing*, vol. 41, no. 12, Dec. 1993.
- [8] Mallat, S. G. “A theory for multiresolution signal decomposition: the wavelet representation”, *IEEE Trans. on PAMI*, vol.11, no.7, 674-693, 1989.
- [9] F. Mokhtarian and A. K. Mackworth, “A theory of multiscale, curvature-based shape representation for planar curves”, *IEEE Trans. on PAMI*, vol.14, no.8, 1992.
- [10] B. A. Olshausen and D. J. Field, “Sparse Coding with an Overcomplete Basis Set: a Strategy Employed by V1?” *Vision Res.*, vol. 37, no. 23 pp.3311-25, 1997.
- [11] T. Sebastina, P. Klein, and B. Kimia, ”Recognition of shapes by editing shock graphs”, *ICCV* 2001.
- [12] K. Siddiqi, K. Tresness, and B. Kimia, ”Parts of visual forms: ecological and psychological aspects”, *Tech. Report. LEMS-104*, Brown University, 1994.
- [13] E.P. Simoncelli, W.T. Freeman, E.H. Adelson, D.J. Heeger, ”Shiftable multi-scale transforms”, *IEEE Transactions on Information Theory*, vol. 38, 587-607, 1992.
- [14] L. H. Staib and J. S. Duncan, ”Boundary finding with parametrically deformable models”, *IEEE Trans. on PAMI*, Nov. 1992.
- [15] S. C. Zhu, C.E. Guo, Y.N. Wu, and Y. Wang, “What are Textons?” *Proc. of the 7th European Conference on Computer Vision*, May 2002.
- [16] S.C. Zhu and A.L. Yuille, “FORMS: a Flexible Object Recognition and Modeling System”, *IJCV*, 20(3), 187-212,1996.