

Wavelet, Active Basis, and Shape Script — A Tour in the Sparse Land

Zhangzhang Si^{*}
UCLA Department of Statistics
Los Angeles, California
zzsi@stat.ucla.edu

Ying Nian Wu[†]
UCLA Department of Statistics
Los Angeles, California
ywu@stat.ucla.edu

ABSTRACT

Sparse coding is a key principle that underlies wavelet representation of natural images. In this paper, we explain that the effort of seeking a common wavelet sparse coding of images from the same object category leads to an *active basis* model, where the images share the same set of selected wavelet elements, which form a linear basis for representing the images. The selected wavelet elements are allowed to perturb their locations and orientations to account for shape deformations, so that the basis becomes active, and the active basis serves as a mathematical representation of a deformable template. We show that a recursive application of the strategy underlying the active basis model leads to a *shape script* model, which is a composition of *shape motifs* such as ellipsoids, parallel bars, angles, etc. These shape motifs are allowed to change their locations, orientations, scales and aspect ratios, and the shape motifs themselves are modeled by active bases. Compared to the active basis model, the shape script model is a sparser representation and therefore has stronger generalization power. It can also be considered another layer of sparse coding of the selected wavelet elements that themselves provide sparse coding of the image intensities.

Categories and Subject Descriptors

I.4.10 [Image Processing and Computer Vision]: Image Representation—*Hierarchical, Statistical*

General Terms

Algorithms

Keywords

Deformable template, Generative model, Sparse coding.

^{*}Zhangzhang Si is a Ph.D. student in UCLA Department of Statistics.

[†]Ying Nian Wu is a professor in UCLA Department of Statistics.

1. INTRODUCTION

Wavelet sparse coding. Wavelet representation has proven to be immensely useful for image analysis and processing. The key principle that underlies wavelet representation is sparsity. That is, an image can typically be represented by a linear superposition of a small number of wavelet elements selected from an appropriate dictionary of wavelet elements. As proposed by Olshausen and Field [12], such a sparse coding strategy may also be employed by the primary visual cortex or V1 for representing retina images. The so-called simple cells in V1 form a dictionary of representational elements. For each retina image, these simple cells compete to explain away the image intensities. By enforcing the sparsity of the representation, Olshausen and Field [12] were able to learn from natural image patches a dictionary of elongated wavelet elements tuned to different locations, orientations and scales. These wavelet elements resemble Gabor wavelets that have been proposed as mathematical models for simple V1 cells [3].

Object template. The discovery of Olshausen and Field [12] naturally begs the question: what is the purpose of wavelet sparse coding in V1, and what is beyond wavelet sparse coding? We argue that a simple excise may provide an answer to this question. Instead of pursuing wavelet sparse coding for generic natural images, we may consider what happens if we pursue a wavelet sparse coding for the image patches of visual objects from the same category, such as images of horses, or images of birds, etc. To start with, we may assume that the objects in these images are roughly aligned, so that they appear at roughly the same locations, scales and poses in these images. Then we may pursue a common wavelet sparse coding simultaneously for these images, so that these images share the same set of wavelet elements. These wavelet elements form a common template of the objects, where each element is like a “stroke” that sketches the template. Mathematically, these wavelet elements form a linear basis for generating the images.

Active basis. In order to account for shape deformations of objects, we may allow the wavelet elements to perturb their locations and orientations before they are linear combined to code each individual image. With such perturbations, the linear basis becomes what we call the *active basis*, which serves as a mathematical representation of deformable template [19]. The active basis template can be learned from a small number of training images. The learned template can then be used to recognize similar objects from testing images by template matching. In both template learning and template matching, there is a local maximization step that

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

MIR’10, March 29–31, 2010, Philadelphia, Pennsylvania, USA.

Copyright 2010 ACM 978-1-60558-815-5/10/03 ...\$10.00.

estimates the optimal perturbation of each selected wavelet element to represent each training or testing image. Such local maximum pooling has been proposed by Riesenhuber and Poggio [13] as the function of the complex cells in V1. In the context of the active basis model, such local maximum pooling serves to deform the active basis template to match the image. The computation of template matching can be implemented by a cortex-like structure of sum-max maps. Such a structure is a variation of the structure proposed by Riesenhuber and Poggio [13].

Shape script. By a recursion of the strategy that underlies the active basis model, we can further generalize it to what we call a *shape script* model. A shape script model is a linear composition of a small number of what we call *shape motifs* selected from a dictionary of motifs. The shape motifs are simple geometric shapes such as ellipsoids, parallel bars, angles, etc. In the shape script model, we may allow the constituent shape motifs to change their locations, orientations, sales, and aspect ratios. Each shape motif is modeled by an active basis, which consists of a small number of Gabor wavelet elements that can change their locations and orientations. Such a shape script model is a highly symbolic and sparse representation of object shapes. We may allow big changes in the parameters of the constituent shape motifs, so that the shape script model has stronger generalization power than the active basis model. The matching of a shape script template to a testing image can be accomplished by a cortex-like structure of recursive sum-max maps, which again is a variation of the structure proposed by Riesenhuber and Poggio [13].

Another layer of sparse coding. The shape script model can be considered another layer of sparse coding on top of wavelet sparse coding. The wavelet sparse coding represents an image in terms of a small number of wavelet elements at different locations and orientation. The shape script model further codes these locations and orientations by a small number of shape motifs of elementary geometric shapes. The shape motifs are represented by active bases, where the activities code the residuals in the locations and orientations. In an analogy to language, if the wavelet elements are letters, then the shape motifs are words.

Relationship with our previous papers. This article is a follow-up to our recent papers [14] [18]. It reports new experimental results on active basis model. The part on the shape script model with shape motifs is new.

2. FROM WAVELET SPARSE CODING TO ACTIVE BASIS

2.1 Wavelet sparse coding

Linear additive model. Let $(\mathbf{I}(x), x = (x_1, x_2) \in D)$ be an image defined on a rectangular lattice or domain D , where $x = (x_1, x_2)$ indexes the pixels of \mathbf{I} . The linear additive model is of the following form:

$$\mathbf{I}(x) = \sum_{i=1}^n c_i B_i(x) + U(x), \quad (1)$$

where $(B_i(x), i = 1, \dots, n)$ are a small number of basis elements selected from a dictionary of such elements, $(c_i, i = 1, \dots, n)$ are the coefficients, and $U(x)$ is the unexplained residual image.

Dictionary of basis elements. The basis elements $(B_i, i = 1, \dots, n)$ are selected from a large dictionary of basis elements. In our work, the basis elements are localized, elongated, and oriented Gabor wavelets [3].

The Gabor wavelets are translated, rotated, and dilated versions of the following function: $G(x_1, x_2) \propto \exp\{-[(x_1/\sigma_1)^2 + (x_2/\sigma_2)^2]/2\}e^{ix_1}$, which is sine-cosine wave multiplied by a Gaussian function. This Gaussian function is elongated along the x_2 -axis, with $\sigma_2 > \sigma_1$, and the sine-cosine wave propagates along the shorter x_1 -axis. We truncate the function to make it locally supported on a finite rectangular range, so that it has a well defined length and width, and the function is 0 outside this rectangular range.

We can translate, rotate and dilate $G(x_1, x_2)$ to obtain a general form of a Gabor wavelet, $B_{x_1, x_2, s, \alpha}$, located at $x = (x_1, x_2)$ and tuned to orientation α and scale s . $B_{x, s, \alpha} = (B_{x, s, \alpha, 0}, B_{x, s, \alpha, 1})$, where $B_{x, s, \alpha, 0}$ is the even-symmetric Gabor cosine component, and $B_{x, s, \alpha, 1}$ is the odd-symmetric Gabor sine component. We always use Gabor wavelets as pairs of cosine and sine components. We normalize both the Gabor sine and cosine components to have zero mean and unit ℓ_2 norm.

For an image $\mathbf{I}(x)$, with $x \in D$, we can project it onto a Gabor wavelet $B_{x, s, \alpha, \eta}$, $\eta = 0, 1$. The projection of \mathbf{I} onto $B_{x, s, \alpha, \eta}$, or the Gabor filter response at (x, s, α) , is $\langle \mathbf{I}, B_{x, s, \alpha, \eta} \rangle = \sum_{x'} \mathbf{I}(x') B_{x, s, \alpha, \eta}(x')$. We write $(\mathbf{I}, B_{x, s, \alpha}) = (\langle \mathbf{I}, B_{x, s, \alpha, 0} \rangle, \langle \mathbf{I}, B_{x, s, \alpha, 1} \rangle)$. The local energy is $|\langle \mathbf{I}, B_{x, s, \alpha} \rangle|^2 = \langle \mathbf{I}, B_{x, s, \alpha, 0} \rangle^2 + \langle \mathbf{I}, B_{x, s, \alpha, 1} \rangle^2$.

The dictionary of Gabor wavelets is $\Omega = \{B_{x, s, \alpha}, \forall (x, s, \alpha)\}$. We can discretize the orientation so that $\alpha \in \{o\pi/O, o = 0, \dots, O-1\}$, that is, O equally spaced orientations (e.g., O is 15 or 16 in our experiments).

Over-completeness and sparsity. The dictionary Ω is called “over-complete” because the number of wavelet elements in Ω is larger than the number of pixels in the image domain, since at each pixel x , there can be many wavelet elements $B_{x, s, \alpha}$ tuned to different orientations α and scales s .

For an image $(\mathbf{I}(x), x \in D)$ we seek to represent it by

$$\mathbf{I}(x) = \sum_{i=1}^n c_i B_{x_i, s, \alpha_i}(x) + U(x), \quad (2)$$

where, corresponding to Equation (1), $B_i(x) = B_{x_i, s, \alpha_i}(x)$, and $(B_{x_i, s, \alpha_i}, i = 1, \dots, n) \subset \Omega$ is a set of Gabor wavelet elements selected from the dictionary Ω . In the experiments described in this paper, we mostly fix the scale parameter s (e.g., the length of the Gabor wavelets is 17 pixels). Even though each B_{x_i, s, α_i} has the same size as the image \mathbf{I} , B_{x_i, s, α_i} is non-zero only on a small rectangular support, so we may consider each B_{x_i, s, α_i} to be a “stroke” for “sketching” the image \mathbf{I} . Thus the linear additive model (2) translates an image with a large number (e.g., 100×100) of pixels into a sketch of a small number (e.g., 50) of strokes, and these strokes capture geometric information in image \mathbf{I} .

Variable selection. The selection of B_{x_i, s, α_i} from the over-complete dictionary Ω and the estimation of c_i is the familiar variable selection problem in linear regression. The set of wavelet elements $\mathbf{B} = (B_{x_i, s, \alpha_i}, i = 1, \dots, n)$ can be selected from Ω by the matching pursuit algorithm [11], which seeks to minimize $\|\mathbf{I} - \sum_{i=1}^n c_i B_{x_i, s, \alpha_i}\|^2$ by a greedy scheme.

0 Initialize $i \leftarrow 0$, $U \leftarrow \mathbf{I}$.

1 Let $i \leftarrow i + 1$. Let $(x_i, \alpha_i) = \arg \max_{x, \alpha} |\langle U, B_{x, s, \alpha} \rangle|^2$.

2 Let $c_i = \langle U, B_{x_i, s, \alpha_i} \rangle$. Update $U \leftarrow U - c_i B_{x_i, s, \alpha_i}$.

3 Stop if $i = n$, else go back to 1.

Currently we hand pick n . n can be selected by principled criteria.

Primary visual cortex. Why do we use a dictionary of localized, elongated and oriented wavelet elements such as Gabor wavelets as representational units? The answer is that they give sparse coding of natural images, which contain edges, and edges can be efficiently represented by such wavelets. A formal mathematical justification is given by the work of Donoho and Candes (1999) on curvelets [2]. A statistical justification is given Olshausen and Field (1996) [12]. They collect a large sample of image patches of natural scenes, and then learn the dictionary of basis elements by minimizing a lasso-like criterion [15] over both the coefficients and the basis elements. The learned basis elements closely resemble the Gabor wavelets. Olshausen and Field (1996) propose that sparse coding in the form of the model (2) is used by the primary visual cortex, where the basis elements $\{B_{x, s, \alpha}\}$ corresponds to the simple cells in primary visual cortex.

The question is, what is the purpose of sparse coding and what is beyond model (2)?

2.2 Active basis model

A simple exercise may offer an answer to the above question. Suppose we want to represent image patches of objects of the same category, and for simplicity let us assume for the present that these image patches are defined on the same lattice, and the objects in these images appear at roughly the same location, scale, and pose. See, for instance, the three deer images in Figure (1.b). Then let us consider what happens if we pursue a sparse coding for these images simultaneously.

Multiple images sharing common basis elements. To fix notation, let $\{\mathbf{I}_m, m = 1, \dots, M\}$ be the set of training images defined on a common lattice D . Since the objects in $\{\mathbf{I}_m\}$ are from the same category, we may want to represent them by $\mathbf{I}_m = \sum_{i=1}^n c_{m,i} B_{x_i, s, \alpha_i} + U_m$, where the multiple images $\{\mathbf{I}_m\}$ share the same set of basis elements $\mathbf{B} = (B_{x_i, s, \alpha_i}, i = 1, \dots, n)$. This \mathbf{B} can be considered a common template for the training images. Because there can be shape deformations in the objects, we may allow the basis elements in \mathbf{B} to perturb their locations and orientations. We call such \mathbf{B} an active basis, which is a mathematical model for a deformable template [19]. The model then becomes

$$\mathbf{I}_m = \sum_{i=1}^n c_{m,i} B_{x_i + \Delta x_{m,i}, s, \alpha_i + \Delta \alpha_{m,i}} + U_m, \quad m = 1, \dots, M. \quad (3)$$

For each image \mathbf{I}_m , the wavelet element B_{x_i, s, α_i} is perturbed to $B_{x_i + \Delta x_{m,i}, s, \alpha_i + \Delta \alpha_{m,i}}$, where $\Delta x_{m,i}$ is the perturbation in location, and $\Delta \alpha_{m,i}$ is the perturbation in orientation. $\mathbf{B}_m = (B_{x_i + \Delta x_{m,i}, s, \alpha_i + \Delta \alpha_{m,i}}, i = 1, \dots, n)$ is the deformed template for representing image \mathbf{I}_m .

Sparse coding for generalization. We call $(\Delta x_{m,i}, \Delta \alpha_{m,i}, i = 1, \dots, n)$ the activities or perturbations of the basis elements for image m . The sparse coding in terms of Gabor wavelets enables us to generalize to similar shapes by perturbing the parameters of the Gabor wavelets, i.e., locations, orienta-

tions, and coefficients. Let

$$A(\alpha) = \{(\Delta x = (d \cos \alpha, d \sin \alpha), \Delta \alpha) : d \in [-b_1, b_1], \Delta \alpha \in [-b_2, b_2]\}$$

be the set of all possible activities for a basis element tuned to orientation α (e.g., $b_1 = 3$ pixels, and $b_2 = \pi/15$). Figure (1.a) illustrates an active basis template of a deer, where each basis element is illustrated by an elongated ellipsoid.

Density substitution. To simplify notation, let $x_{m,i} = x_i + \Delta x_{m,i}$, and $\alpha_{m,i} = \alpha_i + \Delta \alpha_{m,i}$, and $B_{m,i} = B_{x_{m,i}, s, \alpha_{m,i}}$. We can write the deformed template $\mathbf{B}_m = (B_{m,i}, i = 1, \dots, n)$. For simplicity, we assume that the basis elements in the deformed template $(B_{m,i}, i = 1, \dots, n)$ are orthogonal to each other (in practice, we allow small overlap). This is often the case for the linear representation produced by the matching pursuit algorithm. Then $c_{m,i} = \langle \mathbf{I}_m, B_{m,i} \rangle$, and U_m lies in the subspace that is orthogonal to \mathbf{B}_m . Let $C_m = (c_{m,i}, i = 1, \dots, n)$, and with slight abuse of notation, we also use U_m to denote the coordinate of \mathbf{I} in the subspace orthogonal to \mathbf{B}_m . Then $p(\mathbf{I}_m | \mathbf{B}_m) = p(C_m)p(U_m | C_m)$, where the linear mapping between \mathbf{I}_m and (C_m, U_m) is orthogonal. $p(C_m)$ can be estimated from the training images. To model $p(U_m | C_m)$, we introduce a reference model $q(\mathbf{I})$, so that $q(\mathbf{I}_m) = q(C_m)q(U_m | C_m)$ under the same linear mapping. We assume that $p(U_m | C_m) = q(U_m | C_m)$. Then $p(\mathbf{I}_m | \mathbf{B}_m) = q(\mathbf{I}_m)p(C_m)/q(C_m)$. This is a density substitution scheme that has been used by Friedman (1987) [6] for projection pursuit density estimation. Such a scheme also works if C_m is non-orthogonal, or non-linear, or a discrete reduction of \mathbf{I}_m [18].

We assume a $q(\mathbf{I})$ that reproduces the marginal distribution of $c = \langle \mathbf{I}, B_{x, s, \alpha} \rangle$ in natural images, while maintaining the independence of $(c_{m,i}, i = 1, \dots, n)$ for orthogonal \mathbf{B}_m , like the Gaussian white noise model. Let $q(c)$ be this marginal distribution, which can be pooled from natural images, such as the two images of rural and urban scenes in Figure (1). Under $q(c)$, $r = |c|^2$ has a very long tail, reflecting the fact that there are strong edges in natural images or residual background U_m . We then further assume that given \mathbf{B}_m , $(c_{m,i}, i = 1, \dots, n)$ are also independent under $p(C_m)$. This gives us the following model:

$$p(\mathbf{I}_m | \mathbf{B}_m) = q(\mathbf{I}_m) \prod_{i=1}^n \frac{p_i(c_{m,i})}{q(c_{m,i})}. \quad (4)$$

Figure (1.b) illustrates this idea, where p_i and q in this figure are the distributions of $r = |c|^2$ under $p_i(c)$ and $q(c)$ respectively.

Exponential tilting. We further parametrize $p_i(c)$ to be the following exponential family distribution:

$$p(c; \lambda) = \frac{1}{Z(\lambda)} \exp\{\lambda h(|c|^2)\} q(c), \quad (5)$$

where $\lambda > 0$ is the parameter. Let $r = |c|^2$,

$$Z(\lambda) = \int \exp\{\lambda h(r)\} q(c) dc = E_q[\exp\{\lambda h(r)\}]$$

is the normalizing constant, and $\mu(\lambda) = E_\lambda[h(r)]$ is the mean parameter. $h(r)$ is a monotone increasing function. We assume $p_i(c) = p(c; \lambda_i)$.

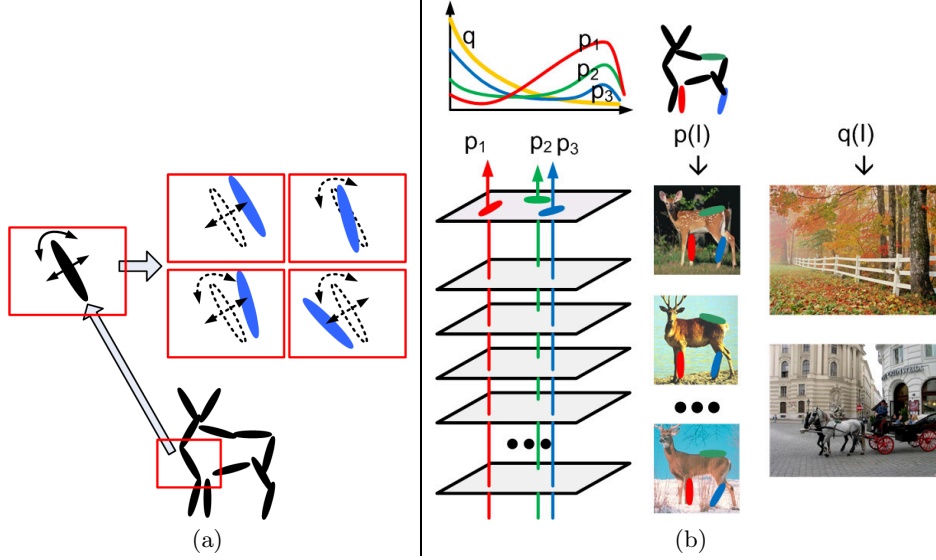


Figure 1: (a) An active basis template $\mathbf{B} = (B_{x_i, s, \alpha_i}, i = 1, \dots, n)$ of a deer, where each Gabor wavelet element B_{x_i, s, α_i} is illustrated by an elongated ellipsoid. An element B_{x_i, s, α_i} (black ellipsoid) can slightly shift its location and orientation and change to $B_{x_i + \Delta x_{m,i}, s, \alpha_i + \Delta \alpha_{m,i}}$ (blue ellipsoids) for coding image \mathbf{I}_m . (b) The elements of the active basis \mathbf{B} are shared by all the training images $\{\mathbf{I}_m, m = 1, \dots, M\}$ of deer, subject to local perturbations $(\Delta x_{m,i}, \Delta \alpha_{m,i}, i = 1, \dots, n)$ that deform the active basis template \mathbf{B} . The elements are selected in the order of the Kullback-Leibler divergence between the foreground distribution p_i of the Gabor filter responses pooled from training images of deer, and the background distribution q pooled from the two natural images of rural and urban scenes.

Sufficient statistics. We use the following function for the sufficient statistics $h(r)$:

$$h(r) = \xi \left(\frac{2}{1 + e^{-2r/\xi}} - 1 \right). \quad (6)$$

$h(r)$ behaves like $h(r) = r$ for small r , but $h(r) \rightarrow \xi$ (e.g., $\xi = 6$) as $r \rightarrow \infty$, i.e., $h(r)$ saturates at ξ . See [18] for a statistical justification of the saturation effect.

Local normalization. For each filter response $|\langle \mathbf{I}_m, B_{x, s, \alpha} \rangle|^2$, we need to normalize it by dividing it by the average response within a local window.

2.3 Learning and inference algorithms

Learning the active basis template from training images. The learning algorithm is essentially a combination of matching pursuit [11] and projection pursuit [6]. We want to pursue the common template \mathbf{B} and deform it to \mathbf{B}_m for each \mathbf{I}_m , so that there is a big contrast between the distribution of $\{c_{m,i}, m = 1, \dots, M\}$ and the marginal distribution $q(c)$. This contrast, or more specifically, the Kullback-Leibler divergence, is monotone in $\sum_{m=1}^M h(|c_{m,i}|^2)$, which serves as the pursuit index. This index essentially counts the number of edges sketched by B_i .

0 Initialize $i \leftarrow 0$. For $m = 1, \dots, M$, initialize $R_m(x, \alpha) \leftarrow \langle \mathbf{I}_m, B_{x, s, \alpha} \rangle$ for all (x, α) .

1 $i \leftarrow i + 1$. Select

$$(x_i, \alpha_i) = \arg \max_{x, \alpha} \sum_{m=1}^M \max_{(\Delta x, \Delta \alpha) \in A(\alpha)} h(|R_m(x + \Delta x, \alpha + \Delta \alpha)|^2).$$

2 For $m = 1, \dots, M$, retrieve

$$(\Delta x_{m,i}, \Delta \alpha_{m,i}) = \arg \max_{(\Delta x, \Delta \alpha) \in A(\alpha_i)} |R_m(x_i + \Delta x, \alpha_i + \Delta \alpha)|^2.$$

Let $c_{m,i} \leftarrow R_m(x_i + \Delta x_{m,i}, \alpha_i + \Delta \alpha_{m,i})$, and update $R_m(x, \alpha) \leftarrow 0$ if

$$|\langle B_{x, s, \alpha}, B_{x_i + \Delta x_{m,i}, s, \alpha_i + \Delta \alpha_{m,i}} \rangle|^2 > \epsilon.$$

Then estimate $\hat{\lambda}_i = \mu^{-1}(\sum_{m=1}^M h(|c_{m,i}|^2)/M)$.

3 Stop if $i = n$, else go back to 1.

See Figure (1) for illustration. We allow small overlap or correlation between $(B_{m,i}, i = 1, \dots, n)$ (e.g., $\epsilon = .1$). The maximum likelihood estimation of λ_i only involves translating the mean parameter back to the natural parameter by $\mu^{-1}()$.

Figure (2) illustrates the results of the learning algorithm. Figure (3) illustrates the learning process.

Matching the active basis template to testing images. After learning the template $\mathbf{B} = (B_{x_i, s, \alpha_i}, i = 1, \dots, n)$ and estimating $\Lambda = (\lambda_i, i = 1, \dots, n)$, we can use the learned deformable template to find the object in a testing image \mathbf{I} , by fitting the following model:

$$\mathbf{I} = \sum_{i=1}^n c_i B_{x + x_i + \Delta x_i, s, \alpha_i + \Delta \alpha_i} + U,$$

where the location of the object, x , is unknown. The maximum likelihood estimation of the location of the object is accomplished by the following inference algorithm:

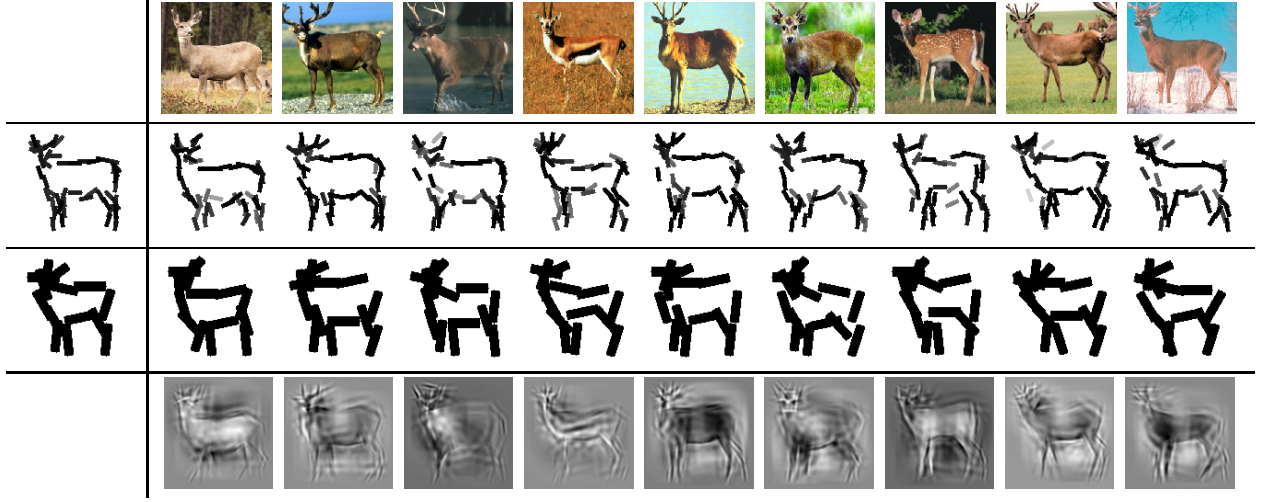


Figure 2: Learning the active basis model. Each Gabor wavelet element is illustrated by a bar with the same location, orientation, and length as the element. The first row displays $\{I_m, m = 1, \dots, M = 9\}$. The second row: the first plot is the active basis template $B = (B_i = B_{x_i, s, \alpha_i}, i = 1, \dots, n = 50)$. The rest of the plots are the deformed templates $B_m = (B_{m,i} = B_{x_i + \Delta x_{m,i}, s, \alpha_i + \Delta \alpha_{m,i}})$. The third row: the same as the second row, except that s is about twice as large, and $n = 14$. The last row displays the linear reconstruction $I_m^{\text{syn}} = \sum_{i=1}^n c_{m,i} B_{m,i}$, where $n = 100$, and $(B_{m,i}, i = 1, \dots, n)$ contains Gabor wavelet elements and difference of Gaussian elements at multiple scales.

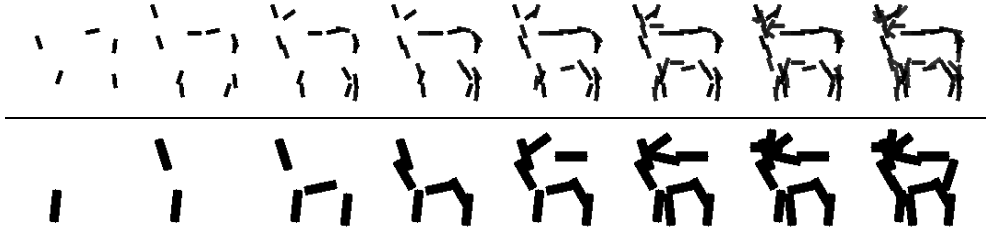


Figure 3: The learning algorithm sequentially selects the elements of the active basis $B = (B_i, i = 1, \dots, n)$. The first row displays the learned template at the smaller scale with $n = 5, 10, 15, 20, 25, 30, 40, 50$. The second row: $n = 1, 2, 4, 6, 8, 10, 12, 14$ at the larger scale.

- 1 For every pixel x , compute the log-likelihood ratio (foreground p versus background q) of x ,

$$l(x) = \sum_{i=1}^n [\lambda_i \max_{(\Delta x, \Delta \alpha) \in A(\alpha_i)} h(|\langle \mathbf{I}, B_{x+x_i+\Delta x, s, \alpha_i+\Delta \alpha} \rangle|^2) - \log Z(\lambda_i)]. \quad (7)$$

- 2 Find the MLE of x : $\hat{x} = \arg \max_x l(x)$. For $i = 1, \dots, n$, retrieve

$$(\Delta x_i, \Delta \alpha_i) = \arg \max_{(\Delta x, \Delta \alpha) \in A(\alpha_i)} |\langle \mathbf{I}, B_{\hat{x}+x_i+\Delta x, s, \alpha_i+\Delta \alpha} \rangle|^2.$$

- 3 Return the location \hat{x} , and the translated and deformed template $(B_{\hat{x}+x_i+\Delta x_i, s, \alpha_i+\Delta \alpha_i}, i = 1, \dots, n)$.

Figure (4) shows two examples of inference. In each example, we search over multiple resolutions of the testing image because the scale of the object in the testing image is unknown. The resolution that achieves the maximum log-likelihood score is selected.

Cortex-like structure. The computation of $l(x)$ in Step 1 of the above inference algorithm can be accomplished by the following three steps:

- 1 For all (x, α) , compute

$$\text{SUM1}(x, \alpha) = h(|\langle \mathbf{I}, B_{x, s, \alpha} \rangle|^2) = h(|\sum_{x'} \mathbf{I}(x') B_{x, s, \alpha}(x')|^2).$$

- 2 For all (x, α) , compute

$$\text{MAX1}(x, \alpha) = \max_{(\Delta x, \Delta \alpha) \in A(\alpha)} \text{SUM1}(x + \Delta x, \alpha + \Delta \alpha).$$

- 3 For all x , compute

$$\text{SUM2}(x) = \sum_{i=1}^n [\lambda_i \text{MAX1}(x + x_i, \alpha_i) - \log Z(\lambda_i)].$$

$$\text{Then } l(x) = [\mathbf{I}, \mathbf{B}_x] = \text{SUM2}(x).$$

These three steps can be implemented by a cortex-like structure, which computes the SUM1 maps, MAX1 maps, and



Figure 4: Inference by template matching. In each block, the left is the testing image I , and the right is the translated and deformed template $(B_{\hat{x}+x_i+\Delta x_i, s, \alpha_i+\Delta \alpha_i}, i = 1, \dots, n)$.

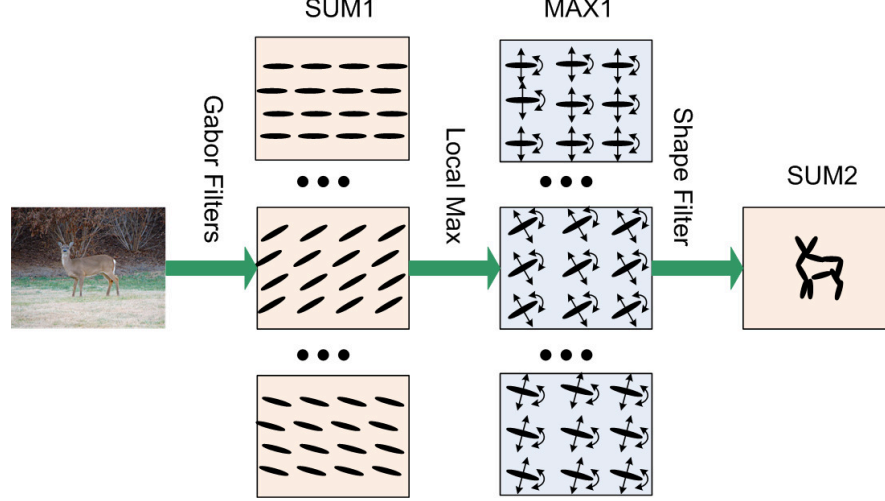


Figure 5: Sum-max maps. The SUM1 maps are obtained by convolving the input image with Gabor filters at all the locations and orientations. The ellipsoids in the SUM1 maps illustrate the local filtering or summation operation. The MAX1 maps are obtained by applying a local maximization operator to the SUM1 maps. The arrows in the MAX1 maps illustrate the perturbations over which the local maximization is taken. The SUM2 map is computed by a summation operator applied to the MAX1 maps, where the summation is over the elements of the active basis, and this summation operator can be interpreted as a shape filter.

SUM2 maps consecutively. See Figure (5). Step 1 corresponds to the simple cells of the primary visual cortex or V1 [12]. Step 2 corresponds to the complex cells of V1 [13]. Step 3 is a consequence of our active basis model. One may hypothesize that it corresponds to cells beyond V1.

The log-likelihood ratio $l(x)$ is the SUM2 score. Writing $\mathbf{B}_x = (B_{x+x_i+\Delta x_i, s, \alpha_i+\Delta \alpha_i}, i = 1, \dots, n)$, we may consider the SUM2 map as a result of convolving I with the deformable “shape filter” \mathbf{B}_x so that $\text{SUM2}(x) = [I, \mathbf{B}_x]$.

2.4 Experiments on learning active basis templates

Supervised learning. The active basis model can be learned from roughly aligned images defined on a common bounding box, where the objects appear at roughly the same location, scale, and pose in the images. Figure (6) displays the learned templates from various training sets.

One can also train the model discriminatively using the adaboost method [7] [17] with weak classifiers of the form $\text{MAX1}_m(x, \alpha)^{1/2} > c$, where the threshold c is to be selected from a grid of 50 equally spaced values at each step. Figure (6) displays the adaboost templates alongside the active basis templates. For each experiment, both templates are

learned from the same positive training set with the same number of elements and under the same parameter setting.

The learning of active basis template does not require negative training images, except a one-dimensional marginal histogram pooled from two background images. It is therefore faster than adaboost. The 1000+ negative image patches for training adaboost templates are randomly cropped from more than 200 large natural images at multiple resolutions. The adaboost learning is initialized from balanced weights, i.e., the total weights for positive images and negative images are both 1/2.

Fitting mixture model by EM algorithm. The training set may be a mixture of different categories or poses. We can fit a mixture of active basis models by the EM algorithm [4], where the M-step learns different active basis templates for different clusters based on the E-step soft classification. Figures (7) and (8) display some examples of EM clustering, initialized by random clustering.

Mixture model and sparse coding. The mixture model may be considered an extreme of sparse coding, where each image patch is coded by a single template selected from a dictionary of multiple templates.

Learning with unknown locations and scales. It is possible to learn the active basis template from non-aligned images

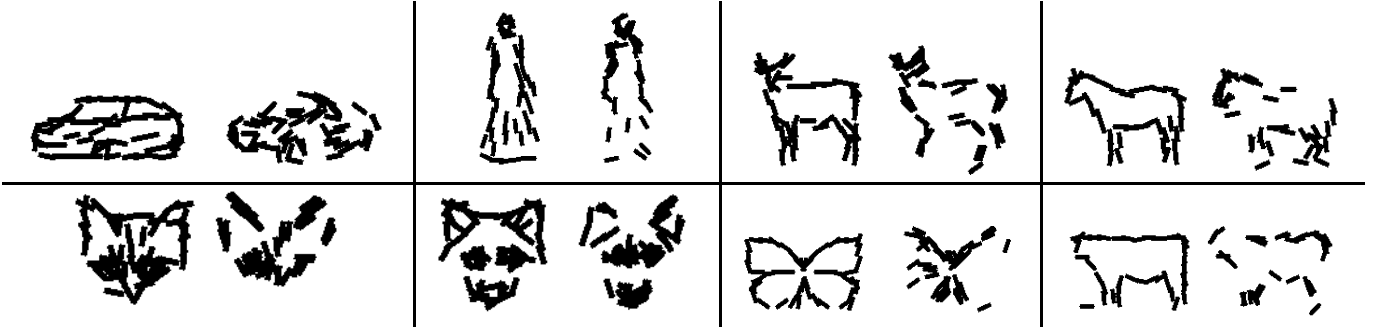


Figure 6: Active basis template (left) and adaboost template (right). Number of elements, number of positives, and number of negatives: Car: 60, 37, 1065. Fashion: 50, 15, 1147. Deer: 50, 9, 1138. Horse: 40, 280, 1511. Cat: 60, 89, 1493. Wolf: 60, 53, 1493. Butterfly: 50, 223, 1004. Cow: 40, 12, 1241.



Figure 7: Fitting mixture model by EM. Number of images: Cat-cattle-wolf-bear: 320. Horse: 188. Fashion: 57.

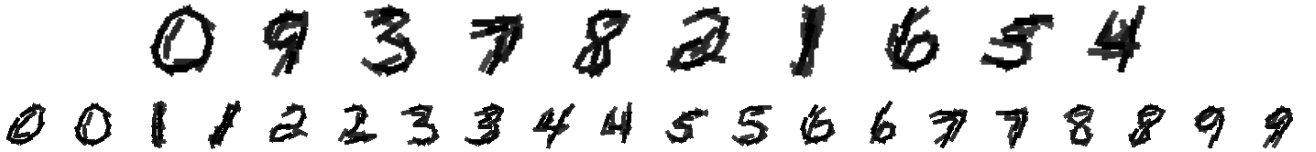


Figure 8: EM mixture. Top row is overall clustering: fitting a mixture model to 500 MNIST [10] images, with the number of clusters set at 10. Bottom row is within digit clustering: fitting a mixture model to 100 or 200 MNIST images from each digit category. Number of clusters is set at 2.

where the objects appear at unknown locations and scales. Figure (9) displays some examples. We initialize the algorithm by learning a template from the first training image. Then we detect the objects in the other images using the learned template. After that we re-learn the template for the detected objects. We iterate the algorithm for a few iterations and then output the final template and the detection results.

3. FROM ACTIVE BASIS TO SHAPE SCRIPT

3.1 Shape script and shape motifs

What is beyond wavelets? The wavelet representations take advantage of the fact that natural images or images of geometric shapes mostly contain edges at different scales. The question is: What is beyond these elements? In an analogy to language, if these elements are “letters,” then what are the “words” so that these “words” lead to even sparser representations?

Artists’ intuition. The question may have already been answered by artists. Figure (10) shows three examples taken

from two recent books on teaching children how to draw animals and other objects by sketching a very small number of elementary geometric shapes [8] [16]. In particular, the first row displays the steps of drawing a horse using an ellipsoid for the body and parallel bars for the legs and so on. The second row displays the drawing of deer and pelican, where for each animal, the first plot illustrates a sparse representation based on elementary shapes. The artists’ intuition is essentially a highly sparse and symbolic representation of animal shapes. We call such a representation the “shape script,” a term coined by Dubinsky and Zhu [5]. A shape script is a highly sparse and symbolic representation that can be useful for learning and inference of object patterns because it captures essential dimensions of the object shapes.

Active basis model for elementary shapes. The shape script can still be described in the linear additive framework, except that the elements are not Gabor wavelets, but elementary shapes, such as ellipsoids, angles, parallel bars, etc. These elementary shapes can be described by active basis templates. We call such elementary active bases “shape motifs,” which are compositions of Gabor wavelet elements.

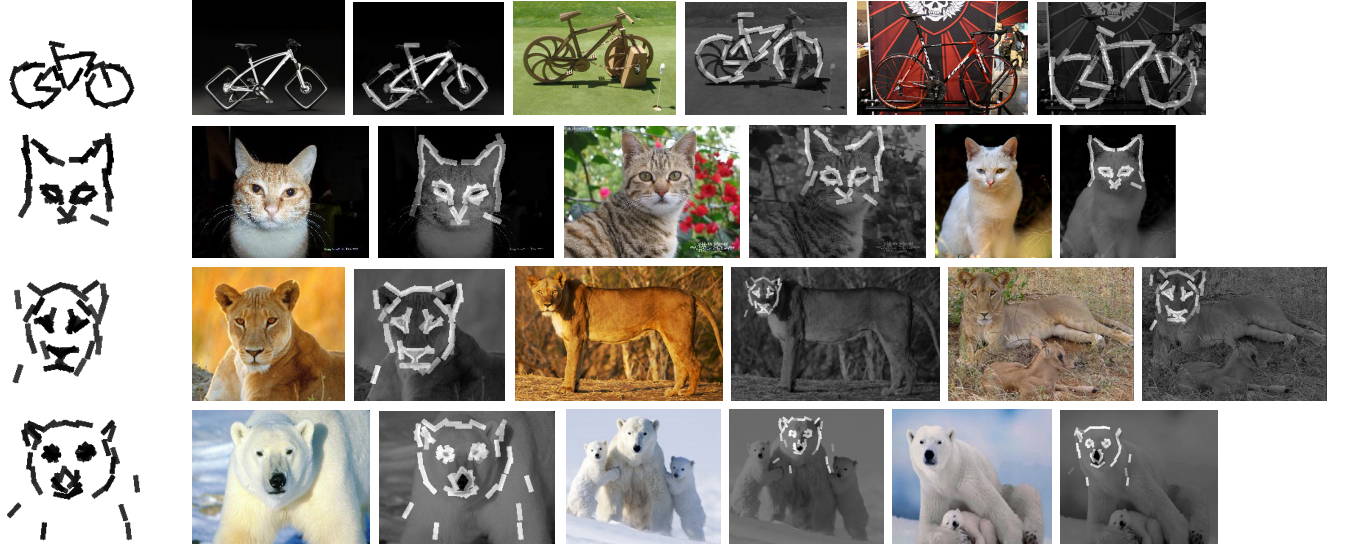


Figure 9: Learning from non-aligned images. Number of training images (3 shown here for each example): Bike: 7. Cat: 9. Lion: 13. Bear: 2.

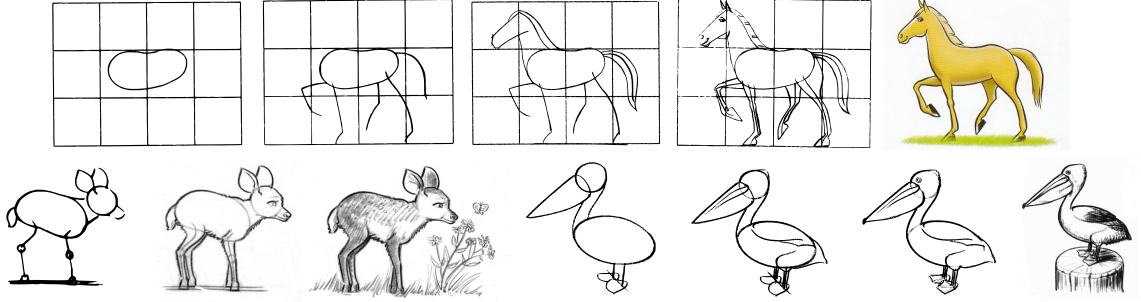


Figure 10: Drawing animals using a very small number of elementary geometric shapes [8][16].

Shape motifs with hyper-parameters. An active basis model can be written in the following form

$$\mathbf{I}_m = \sum_{i=1}^n c_{m,i} B_{x_{m,i}, s_{m,i}, \alpha_{m,i}} + U_m = C_m \mathbf{B}_m + U_m,$$

where \mathbf{B}_m is a deformed template that is composed of Gabor wavelet elements. By a recursion of the above model, we propose the following model that is a composition of K shape motifs which are themselves active basis models:

$$\mathbf{I}_m = \sum_{k=1}^K C_{m,k} \mathbf{B}_{x_{m,k}, s_{m,k}, \rho_{m,k}, \alpha_{m,k}}^{(t_k)} + U_m, \quad (8)$$

where t_k is the type of the k -th shape motif (e.g., ellipsoid, angle, parallel bars, etc.), which is endowed with hyper-parameters, such as the overall location x , scale s , aspect ratio ρ and orientation α . Similar to the active basis, we may allow perturbations of these hyper-parameters, and the perturbations can be quite large because the sizes of the shape motifs are much larger than the Gabor wavelets. Such perturbations cause global deformations of the shape motifs, so that the model is more capable of modeling large deformations and articulations of object shapes. In addition, on top

of the hyper-parameters, we also allow the perturbations of the location, scale, and orientation parameters of the Gabor elements that belong to each shape motif. This causes the local deformations of the shape motifs. So model (9) is a recursive compositional model [9] [21].

3.2 Object recognition by shape script

This subsection illustrates the idea of shape script by a simple experiment of detecting egrets from testing images. We design the shape script template by following the artists' intuition illustrated in Figure (10). The detection process is illustrated in Figure (11) where the shape script template consists of four shape motifs: one ellipsoid for the body, two parallel bars for the neck, and one angle for the beak.

In the current implementation, we assume the following model for each testing image \mathbf{I} :

$$\mathbf{I} = \sum_{k=1}^K C_k \mathbf{B}_{x+x_k, s+s_k, \rho+\rho_k, \alpha+\alpha_k}^{(t_k)} + U,$$

where $K = 4$ is the number of shape motifs, $t_k \in \{\text{ellipsoid, parallel bars, angle}\}$ indexes the type of motif k , $(x_k, s_k, \rho_k, \alpha_k)$ are the location, scale, aspect ratio, and ori-

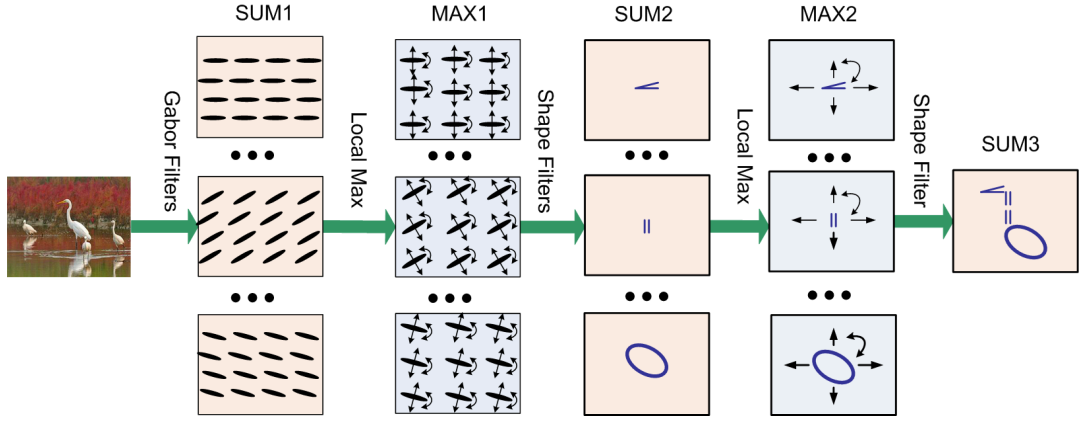


Figure 11: Recursive sum-max maps. The SUM2 maps score the matching of the shape motifs. The MAX2 maps are obtained by local maximum pooling of the SUM2 maps. The SUM3 map scores the matching of the shape script template.

entation of the k -th shape motif. In this article, we design the shape script template by giving the values of the parameters of the four shape motifs. Given the shape script, we will estimate x , the location of the object in the testing image \mathbf{I} , as well as the deformation of the template, i.e., $(\Delta x_k, \Delta s_k, \Delta \rho_k, \Delta \alpha_k, k = 1, \dots, K)$.

The log-likelihood of x is computed by

$$l(x) = \sum_{k=1}^K \max_{(\Delta x, \Delta s, \Delta \rho, \Delta \alpha)} [\mathbf{I}, \mathbf{B}_{x+x_k+\Delta x, s_k+\Delta s, \rho_k+\Delta \rho, \alpha_k+\Delta \alpha}^{(t_k)}], \quad (9)$$

where $[\mathbf{I}, \mathbf{B}_{x+x_k+\Delta x, s_k+\Delta s, \rho_k+\Delta \rho, \alpha_k+\Delta \alpha}^{(t_k)}]$ is computed in the same way as in Equation (7).

Cortex-like structure. Equation (9) can be implemented by a recursive structure of sum-max maps illustrated in Figure (11), which is a recursion of the sum-max maps in Figure (5). In this recursive cortex-like structure, the SUM2 maps score the matching of the shape motifs. The MAX2 maps compute the local maxima of the SUM2 maps. The local maximization computation estimates the shape deformation $(\Delta x_k, \Delta s_k, \Delta \rho_k, \Delta \alpha_k, k = 1, \dots, K)$. The SUM3 map scores the overall matching of the shape script template.

One may hypothesize that the operators for computing the SUM2 maps of the motifs may correspond to neurons beyond V1. These neurons compete to explain away the retina images by the shape motifs.

Bottom-up and top-down. The cortex-like structures in both Figures (11) and (5) are as top-down as they are bottom-up. The bottom-up computation only serves to detect the location of the object. After the detection, a top-down process retrieves the locations, orientations, scales, and aspect ratios of the shape motifs, and then retrieves the locations and orientations of the Gabor wavelet elements of the shape motifs.

Figure (12) displays some examples of detecting egrets by the designed shape script template using the recursive sum-max maps. For each testing image, we superpose the deformed template obtained by the aforementioned top-down retrieval process, where the deformation involves changing the parameters of the shape motifs as well as the parameters of the Gabor wavelet elements of the shape motifs. Because we allow big changes in the parameters of the shape motifs,

the shape script model can account for large deformations, articulations, and pose changes. The SUM3 scores of the detected objects are generally higher than those of natural background images.

The current experiment only serves as a proof of concept. There is still a long way to go to make the model and algorithm robust. For one thing, the current model is a “dislocated” one, where the shape motifs can shift freely. We may need to incorporate “joints” into the model.

The recursive sum-max maps was proposed by Wu, et al. [18] as a variation of the cortex-like structure of Riesenhuber and Poggio [13]. The sum-max maps were also used by Bai, et al. [1] for matching active skeleton template.

We may learn the shape script model from training images using a learning algorithm similar to the one that learns the active basis model. We leave this to future investigation. See also Zhu, et al. [20] on a method for learning hierarchical recursive template.

4. DISCUSSION

The Gabor wavelets provide sparse coding of the image data $\mathbf{I}_m = \sum_{i=1}^n c_{m,i} B_{x_{m,i}, s, \alpha_{m,i}} + U_m$, with residual U_m . The locations and orientations $(x_{m,i}, \alpha_{m,i}, i = 1, \dots, n)$ can be considered the “shape data,” which can be further coded. The active basis model codes the shape data by $(x_{m,i}, \alpha_{m,i}, i = 1, \dots, n) = (x_i, \alpha_i, i = 1, \dots, n) + (\Delta x_{m,i}, \Delta \alpha_{m,i}, i = 1, \dots, n)$, where $(x_i, \alpha_i, i = 1, \dots, n)$ can be considered “mean shape,” and $(\Delta x_{m,i}, \Delta \alpha_{m,i}, i = 1, \dots, n)$ is the residual in coding the shape data, just like U_m is the residual in coding the image data. So the active basis model is a natural tool if we want to code the shape data, because the activities in the active basis model account for the residual in coding the shape data.

The shape script model is to code the shape data by a small number of shape motifs, where each shape motif is an active basis model. So the shape script model is another layer of sparse coding. It would be interesting to learn the shape motifs from natural images or images of specific categories, either by sparse coding or by mixture modeling.

It is worth noting that both the active basis model and the shape script model are built on intensity images directly. There is no need for any pre-processing steps such as edge



Figure 12: Detecting the objects in the testing images using the designed shape script template.

detection or image segmentation to obtain the shape data in the forms of contours or silhouettes. The shape data is obtained simultaneously when we fit the active basis model or the shape script model to the raw intensity image data.

Acknowledgement

We thank Song-Chun Zhu for sharing his ideas on similar models. The work is supported by NSF-DMS 0707055, NSF-IIS 0713652 and Air Force grant FA 9550-08-1-0489.

5. REFERENCES

- [1] X. Bai, X. Wang, W. Liu, L. J. Latecki, and Z. Tu. Active skeleton for non-rigid object detection. In *Proceedings of International Conference on Computer Vision*, 2009.
- [2] E. J. Candes and D. L. Donoho. Curvelets – a surprisingly effective nonadaptive representation for objects with edges. *Curves and Surfaces L. L. Schumaker et al. (eds)*, 1999.
- [3] J. Daugman. Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters. *Journal of Optical Society of America*, 2:1160–1169, 1985.
- [4] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, B*, 39:1–38, 1977.
- [5] A. Dubinsky and S.-C. Zhu. A multiscale generative model for animate shape and parts. In *Proceedings of International Conference on Computer Vision*, 2003.
- [6] J. H. Friedman. Exploratory projection pursuit. *Journal of the American Statistical Association*, 82:249–266, 1987.
- [7] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55:119–139, 1997.
- [8] P. Konye and K. Ashforth. *Funky Things to Draw*. Hinkler Books, 2008.
- [9] S. Geman, D. F. Potter, and Z. Chi. Composition systems. *Quarterly of Applied Mathematics*, 60:707–736, 2002.
- [10] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86:2278–2324, 1998.
- [11] S. Mallat and Z. Zhang. Matching pursuit in a time-frequency dictionary. *IEEE Transactions on Signal Processing*, 41:3397–3415, 1993.
- [12] B. A. Olshausen and D. J. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381:607–609, 1996.
- [13] M. Riesenhuber and T. Poggio. Hierarchical models of object recognition in cortex. *Nature Neuroscience*, 2:1019–1025, 1999.
- [14] Z. Si, H. Gong, S.-C. Zhu, and Y. N. Wu. Learning active basis models by EM-type algorithms. *Statistical Science*, in press, 2009.
- [15] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, B*, 58: 267–288, 1996.
- [16] D. Toll. *You Can Draw: Over 100 Drawings to Master*. Hinkler Books, 2006.
- [17] P. Viola and M. J. Jones. Robust real-time face detection. *International Journal of Computer Vision*, 57:137–154, 2004.
- [18] Y. N. Wu, Z. Si, H. Gong, and S.-C. Zhu. Learning active basis model for object detection and recognition. *International Journal of Computer Vision*, in press, 2009.
- [19] A. L. Yuille, P. W. Hallinan, and D. S. Cohen. Feature extraction from faces using deformable templates. *International Journal of Computer Vision*, 8:99–111, 1992.
- [20] L. Zhu, C. Lin, H. Huang, Y. Chen, and A. Yuille. Unsupervised structure learning: hierarchical recursive composition, suspicious coincidence and competitive exclusion. In *Proceedings of European Conference on Computer Vision*, 2008.
- [21] S.-C. Zhu and D. B. Mumford. A stochastic grammar of images. *Foundations and Trends in Computer Graphics and Vision*, 2:259–362, 2006.