Deformable Template As Active Basis

Ying Nian Wu, Zhangzhang Si, Chuck Fleming, and Song-Chun Zhu
UCLA Department of Statistics
http://www.stat.ucla.edu/∼ywu/ActiveBasis.html

Abstract

This article proposes an active basis model and a shared pursuit algorithm for learning deformable templates from image patches of various object categories. In our generative model, a deformable template is in the form of an active basis, which consists of a small number of Gabor wavelet elements at different locations and orientations. These elements are allowed to slightly perturb their locations and orientations before they are linearly combined to generate each individual training or testing example. The active basis model can be learned from training image patches by the shared pursuit algorithm. The algorithm selects the elements of the active basis sequentially from a dictionary of Gabor wavelets. When an element is selected at each step, the element is shared by all the training examples, in the sense that a perturbed version of this element is added to improve the encoding of each example. Our model and algorithm are developed within a probabilistic framework that naturally embraces wavelet sparse coding and random field.

1. Introduction

1.1. Model, algorithm and theory

The concept of deformable templates [10] is an important element in object recognition. In this article, we present a generative model and a model-based algorithm for learning deformable templates from image patches of various object categories. The machinery we adopt is the wavelet sparse coding model [7] and the matching pursuit algorithm [5]. Our method is a very simple modification of this machinery, with the aim of coding specific ensembles of image patches of various object categories.

We call our model the active basis model, which represents a deformable template in the form of an active basis. An active basis consists of a small number of Gabor wavelet elements at different locations and orientations, and these elements are allowed to slightly perturb their locations and orientations before they are linearly combined to generate each individual training or testing example.

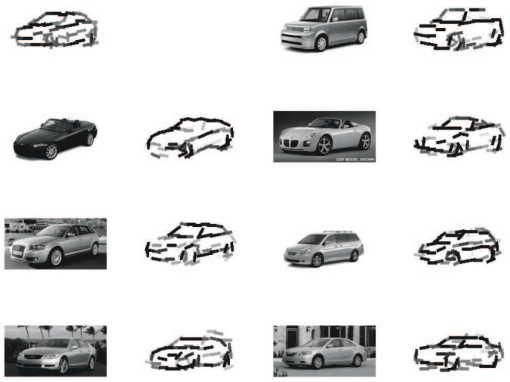

Figure 1. Active basis formed by 60 Gabor wavelet elements. The first plot displays the 60 elements, where each element is represented by a bar. For each of the other 7 pairs, the left plot is the observed image, and the right plot displays the 60 Gabor wavelet elements resulting from locally shifting the 60 elements in the first plot to fit the corresponding observed image.

Figure (1) illustrates the basic idea. It displays 7 image patches of cars at the same scale and in the same pose. These image patches are defined on a common image lattice, which is the bounding box of the cars. These image patches are represented by an active basis consisting of 60 Gabor wavelet elements at different locations and orientation, as displayed in the first plot of figure (1). Each wavelet element is represented symbolically by a bar at the same location and with the same length and orientation. The length of each element is about 1/10 of the length of the image patch. These elements are automatically selected from a dictionary of Gabor wavelet elements at a dense collection of locations and orientations. The selected elements do not have much overlap and are well connected. They form a template of the training image patches.

The 60 elements of the active basis in the first plot are allowed to locally perturb their locations and orientations when they are linearly combined to encode each training or testing example, as illustrated by the remaining 7 pairs of plots of figure (1). For each pair, the left plot displays the observed car image, and the right plot displays the 60

Gabor wavelet elements that are actually used for encoding the corresponding observed image. These 60 elements are perturbed versions of the 60 elements of the active basis displayed in the first plot, so these elements form a deformed template. The deformation of the template is encoded by the local perturbations of the elements of the active basis.

The active basis can be learned from training image patches by a shared pursuit algorithm. The algorithm selects the elements of the active basis sequentially from the dictionary of Gabor wavelets. When an element is selected at each step, the element is shared by all the training examples in the sense that a perturbed version of this element is added to improve the encoding of each example. It is worth noting that for the last two examples in figure (1), the strong edges in the background are not encoded, because these edges are not shared by other examples. Therefore they are ignored by the shared pursuit algorithm.

Our model and algorithm are developed within a theoretical framework that naturally embraces sparse coding and random fields. Specifically, we rewrite the sparse coding model so that the probability distribution of the image intensities can be rigorously defined in terms of tilting a stationary random field by a probability ratio term involving the sparse coding variables.

1.2. Contributions and past work

The contributions of this paper are: (1) An active basis model for representing deformable templates. (2) A shared pursuit algorithm for learning deformable templates. (3) A theoretical framework that integrates sparse coding and random fields.

To credit past work, the active basis model is inspired by the biologically motivated schemes of Riesenhuber and Poggio [8] and Mutch and Lowe [6]. The differences are that we keep track of the deformation of the active basis and maintain the linear additive representation. The shared pursuit algorithm is inspired by the adaboost method of Viola and Jones [9]. The difference is that we work within the framework of generative model. The name "active basis" is clearly derived from "active contours" [4] and "active appearance model." [1] The difference is that our method does not involve control points. Or more precisely, the elements of the active basis play the double role of both control points and linear basis vectors. Lastly, our work is a revision of the texton model [11].

2. Active basis representation

2.1. A dictionary of Gabor wavelets

A Gabor function is of the form: $G(x,y) \propto \exp\{-[(x/\sigma_x)^2 + (y/\sigma_y)^2]/2\}e^{ix}$. We can translate, rotate, and dilate $G(x,y)$ to obtain a general form of Gabor wavelets: $B_{x,y,s,\alpha}(x',y') = G(\tilde{x}/s, \tilde{y}/s)/s^2$, where $\tilde{x} = (x'-x)\cos\alpha - (y'-y)\sin\alpha$, $\tilde{y} = (x'-x)\sin\alpha + (y'-y)\cos\alpha$. s is the scale parameter, and α is the orientation. The central frequency of $B_{x,y,s,\alpha}$ is $\omega = 1/s$.

We normalize the Gabor sine and cosine wavelets to have zero mean and unit l_2 norm. For an image $\mathbf{I}$, the projection coefficient of $\mathbf{I}$ onto $B_{x,y,s,\alpha}$ or the filter response is $\langle \mathbf{I}, B_{x,y,s,\alpha}\rangle = \sum_{x',y'} \mathbf{I}(x',y')B_{x,y,s,\alpha}(x',y')$.

Let $\{(\mathbf{I}_m(x,y), (x,y) \in D), m = 1, ..., M\}$ be a sample of training image patches defined on a domain D of rectangular lattice, and D is the bounding box of the objects of the same category and in the same pose. Our method is scale specific. We fix s so that the length of $B_{x,y,s,\alpha}$ (e.g., 17 pixels) is about 1/10 of the length of D.

The dictionary of Gabor wavelet elements is $\Omega = \{B_{x,y,s,\alpha}, \forall(x,y,s,\alpha)\}$, where (x,y,s,α) are densely discretized: $(x,y) \in D$ with a fine sub-sampling rate (e.g., every 2 pixels), and $\alpha \in \{k\pi/K, k = 0, ..., K-1\}$ (e.g., $K = 15$).

2.2. Active basis

The backbone of the active basis model is

$$\mathbf{I}_m = \sum_{i=1}^{n} c_{m,i} B_{m,i} + \epsilon_m, \tag{1}$$

$$B_{m,i} \approx B_i, \ i = 1, ..., n. \tag{2}$$

where $B_i \in \Omega$, $B_{m,i} \in \Omega$, and $(c_{m,i}, i = 1, ..., n)$ are coefficients. To define $B_{m,i} \approx B_i$, suppose

$$B_i = B_{x_i, y_i, s, \alpha_i}, \tag{3}$$

$$B_{m,i} = B_{x_{m,i}, y_{m,i}, s, \alpha_{m,i}}, \tag{4}$$

then $B_{m,i} \approx B_i$ if and only if there exists $(d_{m,i}, \delta_{m,i})$ such that

$$x_{m,i} = x_i + d_{m,i}\sin\alpha_i, \tag{5}$$

$$y_{m,i} = y_i + d_{m,i}\cos\alpha_i, \tag{6}$$

$$\alpha_{m,i} = \alpha_i + \delta_{m,i}, \tag{7}$$

$$d_{m,i} \in [-b_1, b_1], \ \delta_{m,i} \in [-b_2, b_2]. \tag{8}$$

That is, we allow B_i to shift its location along its normal direction, and we also allow B_i to shift its orientation. b_1 and b_2 are the bounds for the allowed displacement in location and turn in orientation (e.g., $b_1 = 6$ pixels, and $b_2 = \pi/15$).

In the above notation, the deformable template is the active basis $\mathbf{B} = (B_i, i = 1, ..., n)$. The deformed template or the activated basis is $\mathbf{B}_m = (B_{m,i}, i = 1, ..., n) \approx \mathbf{B}$. See figure (1) for illustration.

2.3. Shared matching pursuit for least squares

Given the examples $\{\mathbf{I}_m, m = 1, ..., M\}$, we can learn the template $\mathbf{B}$ and its deformed versions $\{\mathbf{B}_m \approx$

$\mathbf{B}, m = 1, ..., M\}$. We may use the least squares criterion $\sum_{m=1}^{M} \|\mathbf{I}_m - \sum_{i=1}^{n} c_{m,i} B_{m,i}\|^2$ to drive the following shared matching pursuit algorithm.

(0) For $m = 1, ..., M$, let $\epsilon_m \leftarrow \mathbf{I}_m$. Let $i \leftarrow 1$.

(1) For each putative candidate $B_i \in \Omega$, do the following: For $m = 1, ..., M$, choose the optimal $B_{m,i}$ that maximizes $|\langle \epsilon_m, B_{m,i} \rangle|^2$ among all possible $B_{m,i} \approx B_i$. Then choose that particular candidate B_i with the maximum corresponding $\sum_m |\langle \epsilon_m, B_{m,i} \rangle|^2$.

(2) For $m = 1, ..., M$, let $c_{m,i} \leftarrow \langle \epsilon_m, B_{m,i} \rangle$, and let $\epsilon_m \leftarrow \epsilon_m - c_{m,i} B_{m,i}$.

(3) Stop if $i = n$. Otherwise let $i \leftarrow i + 1$, and go to (1).

In this article, we choose to adopt the more general probabilistic formulation, where the least squares criterion is a special case of the log-likelihood.

3. Probabilistic formulation

With the active basis representation (1) and (2) as the backbone, we can put probability distributions on the variables in the representation in order to construct a generative model. With such a model, learning can be based on likelihood.

3.1. Rewriting sparse coding model

Given template $\mathbf{B} = (B_i, i = 1, ..., n)$, we assume that $(d_{m,i}, \delta_{m,i}) \sim \text{uniform}(\Delta = [-b_1, b_1] \times [-b_2, b_2])$ in order to generate the deformed template $\mathbf{B}_m = (B_{m,i}, i = 1, ..., n)$ according to (3)-(8).

Given deformed template $\mathbf{B}_m = (B_{m,i}, i = 1, ..., n)$, we need to specify the distribution of the foreground coefficients $c_m = (c_{m,i}, i = 1, ..., n)$, and the distribution of the background residual ϵ_m, in order to generate $\mathbf{I}_m$ according to (1). The commonly assumed model is

$$(c_{m,1}, ..., c_{m,n}) \sim g(c_{m,1}, ..., c_{m,n}), \tag{9}$$

$$\epsilon_m(x, y) \sim \text{N}(0, \sigma^2) \text{ independently}, \tag{10}$$

$$(c_{m,1}, ..., c_{m,n}) \text{ is independent of } \epsilon_m. \tag{11}$$

There are two problems with the above specification. (1) A white noise model does not capture the texture properties of the background. (2) The foreground distribution g cannot be estimated in closed form because we must deconvolve the additive noise ϵ_m. The following observation helps solve these two problems.

Theorem 1 *For the representation (1), given $\mathbf{B}_m = (B_{m,i}, i = 1, ..., n)$, under the assumptions (9), (10) and (11), the distribution of $\mathbf{I}_m$ is*

$$p(\mathbf{I}_m \mid \mathbf{B}_m) = q(\mathbf{I}_m) \frac{p(r_{m,1}, ..., r_{m,n})}{q(r_{m,1}, ..., r_{m,n})}, \tag{12}$$

where $r_{m,i} = \langle \mathbf{I}_m, B_{m,i} \rangle$, $i = 1, ..., n$. $q(\mathbf{I}_m)$ is the density of white noise model, i.e., $\mathbf{I}_m(x, y) \sim \text{N}(0, \sigma^2)$ independently. $q(r_{m,1}, ..., r_{m,n})$ is the density of $(r_{m,1}, ..., r_{m,n})$ under $q(\mathbf{I}_m)$. $p(r_{m,1}, ..., r_{m,n})$ is the density of $(r_{m,1}, ..., r_{m,n})$ under $p(\mathbf{I}_m \mid \mathbf{B}_m)$.

The proof is given in the appendix. The basic idea of the proof is very simple. By adding $\sum_i c_{m,i} B_{m,i}$ to the white noise background ϵ_m, we only change the dimensions of ϵ_m within the subspace spanned by $(B_{m,i}, i = 1, ..., n)$, without disturbing the rest of the dimensions. This can be accomplished by multiplying $q(\mathbf{I}_m)$ by the probability ratio $p(r_{m,1}, ..., r_{m,n})/q(r_{m,1}, ..., r_{m,n})$, which changes the distribution of $(r_{m,1}, ..., r_{m,n})$ from $q(r_{m,1}, ..., r_{m,n})$ to $p(r_{m,1}, ..., r_{m,n})$, without changing the distribution of the remaining dimensions.

We may use the compact matrix notation. Let $\mathbf{I}_m$ be the $|D| \times 1$ vector, where $|D|$ is the number of pixels in domain D. Let $\mathbf{B}_m = (B_{m,1}, ..., B_{m,n})$ be the $|D| \times n$ matrix, where each column is a vectorized version of $B_{m,i}$. Let $r_m = (r_{m,1}, ..., r_{m,n})'$ be the $n \times 1$ vector of sparse coding variables. Then $r_m = \mathbf{B}_m' \mathbf{I}_m$.

The foreground $p(r_m)$ can be estimated directly by pooling the sample $\{r_m = \mathbf{B}_m' \mathbf{I}_m, m = 1, ..., M\}$, which are responses of Gabor wavelets at fixed locations (subject to local perturbations $\mathbf{B}_m \approx \mathbf{B}$), so we do not need to estimate g, which involves unnecessary deconvolution of the additive noise ϵ_m.

Under the white noise model $q(\mathbf{I}_m)$ where $\mathbf{I}_m(x, y) \sim \text{N}(0, \sigma^2)$ independently, we have $r_m \sim \text{N}(0, \mathbf{B}_m' \mathbf{B}_m \sigma^2)$, so $q(r_m)$ is in closed form.

Log-likelihood and KL-divergence. We can estimate the template $\mathbf{B}$ and its deformed versions $\{\mathbf{B}_m \approx \mathbf{B}, m = 1, ..., M\}$ by maximizing the log-likelihood

$$\sum_{m=1}^{M} \log[p(\mathbf{I}_m \mid \mathbf{B}_m)/q(\mathbf{I}_m)] = \sum_{m=1}^{M} \log \frac{p(r_m)}{q(r_m)}. \tag{13}$$

As $M \to \infty$, the log-likelihood per observation

$$\frac{1}{M} \sum_{m=1}^{M} \log \frac{p(r_m)}{q(r_m)} \to \text{KL}(p(r_m) | q(r_m)), \tag{14}$$

which is the Kullback-Leibler divergence from $p(r_m)$ to $q(r_m)$.

Equivalence to least squares. Under white noise $q(\mathbf{I}_m)$, $r_m = \mathbf{B}_m' \mathbf{I}_m \sim \text{N}(0, \mathbf{B}_m' \mathbf{B}_m \sigma^2)$. If we assume $p(r_m)$ is such that $r_m \sim \text{N}(0, \mathbf{B}_m' \mathbf{B}_m \sigma_0^2)$ with $\sigma_0^2 > \sigma^2$, then $\log[p(r_m)/q(r_m)]$ is positively linear in $r_m' (\mathbf{B}_m' \mathbf{B}_m)^{-1} r_m = \mathbf{I}_m' \mathbf{B}_m (\mathbf{B}_m' \mathbf{B}_m)^{-1} \mathbf{B}_m' \mathbf{I}_m$, which is the squared norm of the projection of $\mathbf{I}_m$ onto the subspace spanned by $\mathbf{B}_m$, which equals to $\|\mathbf{I}_m\|^2 - \min_{c_m} \|\mathbf{I}_m - \mathbf{B}_m c_m\|^2$, where $c_m = (c_{m,1}, ..., c_{m,n})'$. So maximizing

the log-likelihood (13) with such $p(r_m)$ is equivalent to the least squares criterion.

Orthogonality. The norm $\mathbf{I}'_m\mathbf{B}_m(\mathbf{B}'_m\mathbf{B}_m)^{-1}\mathbf{B}'_m\mathbf{I}_m$ naturally favors the selection of orthogonal $\mathbf{B}_m$. In this article, we enforce that $\mathbf{B}'_m\mathbf{B}_m \approx \mathbf{1},\ m = 1, ..., M$, for simplicity, where "$\mathbf{1}$" denotes the identity matrix. That is, we enforce that the elements in the deformed template $\mathbf{B}_m = (B_{m,i}, i = 1, ..., n)$ are approximately orthogonal to each other, or do not have much overlap. The precise definition of $\mathbf{B}'_m\mathbf{B}_m \approx \mathbf{1}$ is: $\langle B_{m,i}, B_{m,j}\rangle < \zeta$ for $i \neq j$, where ζ is a small threshold (e.g., $\zeta = .1$).

Random field tilting. Equation (12) is actually more general than is defined in Theorem 1: (1) The background $q(\mathbf{I}_m)$ can be any random field. (2) The sparse coding variables $(r_{m,i}, i = 1, ..., n)$ can be any deterministic transformations of $\mathbf{I}_m$. In this more general context, (12) is a random field tilting scheme, which consists of (1) Replacing background $q(r_{m,1}, ..., r_{m,n})$ by foreground $p(r_{m,1}, ..., r_{m,n})$. (2) Retaining the conditional distribution of the remaining $|D| - n$ dimensions of $\mathbf{I}_m$ given $(r_{m,1}, ..., r_{m,n})$. The remaining $|D| - n$ dimensions are implicit. This is a generalized version of projection pursuit [3]. The following are some perspectives to view this scheme:

(1) *Hypothesis testing.* $q(\mathbf{I}_m)$ can be considered the null hypothesis. $p(r_{m,1}, ..., r_{m,n})$ can be considered the test statistics to reject $q(\mathbf{I}_m)$. The above scheme modifies the null hypothesis to an alternative hypothesis.

(2) *Classification.* $q(\mathbf{I}_m)$ can be considered the ensemble of negative examples. $p(\mathbf{I}_m)$ is the ensemble of positive examples. The sparse coding variables $(r_{m,i}, i = 1, ..., n)$ are the features that distinguish the two ensembles.

(3) *Coding.* Instead of coding $(r_{m,i}, i = 1, ..., n)$ by q, we code them by p. The gain in coding length is the KL-divergence (14).

3.2. Model specification

Sparse coding variables. Given $\mathbf{B}_m = \{B_{m,i}, i = 1, ..., n\}$, with $\mathbf{B}'_m\mathbf{B}_m \approx \mathbf{1}$, we choose to use $r_{m,i} = h_m(|\langle \mathbf{I}_m, B_{m,i}\rangle|^2),\ i = 1, ..., n$, as sparse coding variables. $|\langle \mathbf{I}_m, B_{m,i}\rangle|^2$ is the local energy, which is the sum of squares of the responses from the pair of Gabor cosine and sine wavelets. We ignore the local phase information, which is unimportant for shapes. $h_m()$ is a monotone normalization transformation that is independent of object categories.

To specify the model, we need to (1) specify the background $q(\mathbf{I}_m)$ and derive $h_m()$ and $q(r_{m,1}, ..., r_{m,n})$. (2) specify the foreground $p(r_{m,1}, ..., r_{m,n})$. Figure (2) illustrates the idea. The shaded rectangles are training images. We can pool these images to estimate $p(r_{m,1}, ..., r_{m,n})$, as illustrated by the vertical arrows at specific locations. $p(r_{m,1}, ..., r_{m,n})$ is to be contrasted against the background $q(r_{m,1}, ..., r_{m,n})$, which is not location specific, as illustrated by the horizontal arrow, because $q(\mathbf{I}_m)$ is stationary. We use the ambiguous notation $p(r)$ and $q(r)$ in figure (2) to mean either the joint distribution of $r_{m,1}, ..., r_{m,n}$ or the marginal distributions of individual components. The template $\mathbf{B}$ and its deformed versions $\{\mathbf{B}_m \approx \mathbf{B}\}$ should be chosen to maximize the KL-divergence from p to q, as dictated by equation (14).

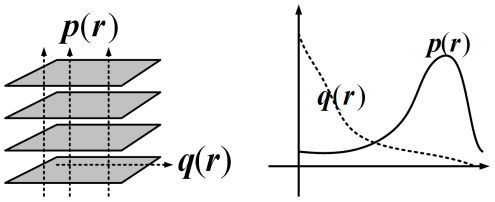

Figure 2. $p(r_{m,1}, ..., r_{m,n})$ is pooled over training images (shaded rectangles) at specific locations. $q(r_{m,1}, ..., r_{m,n})$ is derived from stationary background $q(\mathbf{I}_m)$.

Background model $q(\mathbf{I}_m)$ and $q(r_{m,1}, ..., r_{m,n})$. The most natural $q(\mathbf{I}_m)$ from the classification perspective is the generic ensemble of natural image patches. In the following, we derive $h_m()$ and $q(r_{m,1}, ..., r_{m,n})$ by gradually generalizing from the white noise model.

(1) *White noise* $\mathbf{I}_m(x, y) \sim \mathrm{N}(0, \sigma_m^2)$, where σ_m^2 can be estimated by the marginal variance of $\mathbf{I}_m$. $|\langle \mathbf{I}_m, B_{m,i}\rangle|^2$ is the sum of squares of two independent normal random variables of variance σ_m^2, so $|\langle \mathbf{I}_m, B_{m,i}\rangle|^2 \sim \sigma_m^2\chi_2^2 \sim 2\sigma_m^2 \exp(1)$, i.e., the exponential distribution, and $|\langle \mathbf{I}_m, B_{m,i}\rangle|^2/2\sigma_m^2 \sim \exp(1)$. If $\mathbf{B}'_m\mathbf{B}_m = \mathbf{1}$, then $|\langle \mathbf{I}_m, B_i\rangle|^2/2\sigma_m^2$ are independent for $i = 1, ..., n$.

(2) *Stationary isotropic Gaussian* $q(\mathbf{I}_m)$. Let s be the common scale of $\mathbf{B}_m = (B_{m,i}, i = 1, ..., n)$. $B_{m,i}$ can sense $\mathbf{I}_m$ only within a limited frequency band around $1/s$. Let $\sigma_{m,s}^2 = \mathrm{E}[|\langle \mathbf{I}_m, B_{x,y,s,\alpha}\rangle|^2]$, and assume that the spectrum of the Gaussian process is locally flat within the above-mentioned frequency band, then as far as $\mathbf{B}_m$ can sense, $q(\mathbf{I}_m)$ is no different than white noise $\mathrm{N}(0, \sigma_{m,s}^2/2)$. Therefore, $|\langle \mathbf{I}_m, B_i\rangle|^2/\sigma_{m,s}^2 \sim \exp(1)$, and this is a whitening transformation. $|\langle \mathbf{I}_m, B_i\rangle|^2/\sigma_{m,s}^2$ are independent for $i = 1, ..., n$ if $\mathbf{B}'_m\mathbf{B}_m = \mathbf{1}$. $\sigma_{s,m}^2$ can be estimated by

$$\hat{\sigma}_{m,s}^2 = \frac{1}{|D|K} \sum_{x,y \in D} \sum_{\alpha} |\langle \mathbf{I}_m, B_{x,y,s,\alpha}\rangle|^2, \qquad (15)$$

where K is the total number of orientations. The tail of the distribution is $\Pr(|\langle \mathbf{I}_m, B_i\rangle|^2/\sigma_{m,s}^2 > r) = \exp(-r)$, which is short.

(3) *Generic ensemble of natural image patches.* If we pool the marginal distribution of $|\langle \mathbf{I}_m, B_{m,i}\rangle|^2/\sigma_{m,s}^2$ over the ensemble of natural image patches, and let $F(r) = \Pr(|\langle \mathbf{I}_m, B_{m,i}\rangle|^2/\sigma_{m,s}^2 > r)$ be the tail of this marginal distribution, then $F(r) \gg \exp(-r)$ for large r, because there are strong edges in this ensemble. The transformation that equates the tails $F(r) = \exp(-r_0)$ is $r_0 = -\log F(r)$,

so $-\log F(|\langle \mathbf{I}_m, B_{m,i}\rangle|^2/\sigma^2_{m,s}) \sim \exp(1)$. $-\log F$ is a non-linear whitening transformation. Therefore, we have

$$r_{m,i} = h_m(|\langle \mathbf{I}_m, B_{m,i}\rangle|^2) = -\log F(|\langle \mathbf{I}_m, B_{m,i}\rangle|^2/\sigma^2_{m,s}).$$

We assume that the generic ensemble inherits from Gaussian process the property that $(r_{m,i}, i = 1, ..., n)$ are independent under $\mathbf{B}'_m\mathbf{B}_m = \mathbf{1}$. So $q(r_{m,1}, ..., r_{m,n}) = \exp\{-\sum_{i=1}^n r_{m,i}\}$, i.e., $r_{m,i} \sim \exp(1)$ independently for $i = 1, ..., n$.

One can learn $F(r)$ by the tail proportions in the marginal histogram of natural images. In our current implementation, we use a crude but simple approximation. Because $-\log F(r) \ll r$ for large r, we assume a saturation threshold $\xi > 0$, and approximate $-\log F(r) \approx \min(r, \xi)$ (e.g., $\xi = 16$).

Foreground model $p(r_{m,1}, ..., r_{m,n})$. We assume the simplest model for $p(r_{m,1}, ..., r_{m,n})$: $r_{m,i} \sim \exp(\lambda_i)$ independently for $i = 1, ..., n$, with $\lambda_i < 1$. The density of $r_{m,i}$ is $p(r) = \lambda_i \exp(-\lambda_i r)$. This is the maximum entropy model under the constraint $\mathrm{E}_p(r_{m,i}) = 1/\lambda_i$.

Log-likelihood is

$$\log[p(\mathbf{I}_m \mid \mathbf{B}_m)/q(\mathbf{I}_m)] = \sum_{i=1}^n \log \frac{p(r_{m,i})}{q(r_{m,i})} = \sum_{i=1}^n [(1-\lambda_i)r_{m,i} + \log \lambda_i]. \quad (16)$$

Given $\mathbf{B}$, the prior distribution of $\mathbf{B}_m$ is uniform: $p(\mathbf{B}_m \mid \mathbf{B}) = 1/|\Delta|^n$, where $\Delta = [-b_1, b_1] \times [-b_2, b_2]$ is the allowed range of shifting in location and orientation for each B_i, and $|\Delta|$ is the size of Δ. So the posterior distribution $p(\mathbf{B}_m \mid \mathbf{I}_m, \mathbf{B}) \propto p(\mathbf{I}_m \mid \mathbf{B}_m)/q(\mathbf{I}_m)$. Thus, $B_{m,i}$ can be estimated by maximizing $r_{m,i}$ or $|\langle \mathbf{I}_m, B_{m,i}\rangle|^2$ among all possible $B_{m,i} \approx B_i$, subject to that $(B_{m,i}, i = 1, ..., n)$ are approximately non-overlapping.

Given $\{\mathbf{B}_m, m = 1, ..., M\}$, λ_i can be estimated by pooling $\{r_m = \mathbf{B}'_m\mathbf{I}_m, m = 1, ..., M\}$. The maximum likelihood estimate is $\hat{\lambda}_i = 1/\bar{r}_i$, where $\bar{r}_i = \sum_{m=1}^M r_{m,i}/M$ is the average response. Replacing λ_i by $\hat{\lambda}_i$, the log-likelihood or coding gain per image

$$\frac{1}{M}\sum_{m=1}^M \log[p(\mathbf{I}_m, \mathbf{B}_m \mid \mathbf{B})/q(\mathbf{I}_m)] = \sum_{i=1}^n (\bar{r}_i - 1 - \log \bar{r}_i) - n \log |\Delta|,$$

where $p(\mathbf{I}_m, \mathbf{B}_m \mid \mathbf{B}) = p(\mathbf{I}_m \mid \mathbf{B}_m)p(\mathbf{B}_m \mid \mathbf{B})$. $\log |\Delta|$ is the cost for coding the shifting from B_i to $B_{m,i}$. We can sequentially introduce the elements of $\mathbf{B} = (B_i, i = 1, ..., n)$ by maximizing $\bar{r}_i$ subject to

$$\bar{r}_i - 1 - \log \bar{r}_i > \log |\Delta|, \quad (17)$$

and the elements in $\mathbf{B}_m = (B_{m,i}, i = 1, ..., n)$ are approximately non-overlapping.

3.3. Shared pursuit for maximum likelihood

We use the notation $\partial B_{m,i}$ to denote all the $B \in \Omega$, such that $\langle B, B_{m,i}\rangle > \zeta$, i.e., those elements that overlap with $B_{m,i}$.

(0) For $m = 1, ..., M$, and for each $B \in \Omega$, compute $[\mathbf{I}_m, B] = -\log F(|\langle \mathbf{I}_m, B\rangle|^2/\sigma^2_{m,s})$ with $\sigma^2_{m,s}$ estimated by (15). Set $i \leftarrow 1$.

(1) For each putative candidate $B_i \in \Omega$, do the following: For $m = 1, ..., M$, choose the optimal $B_{m,i}$ that maximizes $[\mathbf{I}_m, B_{m,i}]$ among all possible $B_{m,i} \approx B_i$. Then choose that particular candidate B_i with the maximum corresponding $\sum_m [\mathbf{I}_m, B_{m,i}]$. Set $\lambda_i = M/\sum_m [\mathbf{I}_m, B_{m,i}]$.

(2) For $m = 1, ..., M$, for each $B \in \partial B_{m,i}$, set $[\mathbf{I}_m, B] \leftarrow 0$, to enforce approximate non-overlapping constraint.

(3) Stop if $i = n$. Otherwise let $i \leftarrow i + 1$, and go to (1).

The stopping criterion can also be based on (17).

Find and sketch. We can used the learned model, in particular, $\mathbf{B} = (B_i, i = 1, ..., n)$, and $\lambda = (\lambda_i, i = 1, ..., n)$, to find the object in a new testing image $\mathbf{I}_m$, $m \notin \{1, ..., M\}$. Suppose $\mathbf{I}_m$ is defined on domain D_m, which can be much larger than the bounding box D. We slide D over D_m. Let $D_{x,y} \subset D_m$ be the bounding box centered at $(x, y) \in D_m$. Within each $D_{x,y}$, for $i = 1, ..., n$, choose the optimal $B_{m,i} \approx B_i$ that maximizes $r_{m,i} = [\mathbf{I}_m, B_{m,i}]$. Then compute the log-likelihood score $l_m(x, y) = \sum_{i=1}^n [(1-\lambda_i)r_{m,i} + \log \lambda_i]$. Choose (x, y) with maximum log-likelihood score $l_m(x, y)$. The corresponding $\mathbf{B}_m = (B_{m,i}, i = 1, ..., n)$ is the sketch of the object. If the size of the object in the testing $\mathbf{I}_m$ is different than the size of objects in the training images, we can scale $\mathbf{I}_m$ to obtain a sequence of zoomed versions of $\mathbf{I}_m$. Then we can choose the optimal scale based on the maximum log-likelihood scores obtained over multiple scales.

4. Active mean vector and active correlation

The deformable template $\mathbf{B} = (B_i, i = 1, ..., n)$ in the above section is parametrized by $\lambda = (\lambda_i, i = 1, ..., n)$. The log-likelihood score is $\sum_{i=1}^n [(1-\lambda_i)r_{m,i} + \log \lambda_i]$, which is non-linear in λ. This motivates us to introduce a simpler linear score without explicit probabilistic assumptions.

4.1. Linear scoring

We parametrizes the deformable template $\mathbf{B} = (B_i, i = 1, ..., n)$ by $\theta = (\theta_i, i = 1, ..., n)$, where θ is a unit vector with $\|\theta\|^2 = 1$. We replace the log-likelihood score

Figure 3. The first plot is $\mathbf{B} = \{B_i, i = 1, ..., n\}$, $n = 48$, where each B_i is represented by a bar. For the rest $M = 37$ examples, the left is $\mathbf{I}_m$, and the right is $\mathbf{B}_m = (B_{m,i}, i = 1, ..., n)$. The M examples are listed in the descending order of log-likelihood.

by $\langle \theta, r_m^{1/2} \rangle = \sum_{i=1}^n \theta_i r_{m,i}^{1/2}$, where $r_m^{1/2} = (r_{m,i}^{1/2}, i = 1, ..., n)$. Given $\mathbf{B}$, $\mathbf{B}_m = (B_{m,i} \approx B_i, i = 1, ..., n)$ can be chosen by maximizing $\langle \theta, r_m^{1/2} \rangle$, subject to the approximate non-overlapping constraint. We call this maximum the active correlation, which filters out local deformation as well as local phase information. $\mathbf{B}$ and θ can be estimated by maximizing $\sum_{m=1}^M \langle \theta, r_m^{1/2} \rangle$. θ is the mean vector in the active basis. If $M = 2$, the maximum of $\sum_{i=1}^n (r_{1,i} r_{2,i})^{1/2}$ is the pairwise active correlation between $\mathbf{I}_1$ and $\mathbf{I}_2$.

4.2. Shared pursuit for maximum correlation

(0) The same as maximum likelihood.

(1) For each putative candidate $B_i \in \Omega$, do the following: For $m = 1, ..., M$, choose the optimal $B_{m,i}$ that maximizes $[\mathbf{I}_m, B_{m,i}]$ among all possible $B_{m,i} \approx B_i$. Then choose that particular candidate B_i with the maximum corresponding $\sum_m [\mathbf{I}_m, B_{m,i}]^{1/2}$. Set $\theta_i = \sum_m [\mathbf{I}_m, B_{m,i}]^{1/2} / M$.

(2) The same as maximum likelihood.

(3) If $i = n$, normalize θ so that $\|\theta\|^2 = 1$, then stop. Otherwise let $i \leftarrow i + 1$, and go to (1).

We can also use the active correlation score for find-and-sketch.

5. Experiments

Parameter values. Size of Gabor wavelets = 17×17. (x, y) is sub-sampled every 2 pixels. The orientation α takes $K = 15$ equally spaced angles in $[0, \pi]$. The saturation threshold in approximation $-\log F(r) \approx \min(r, \xi)$ is $\xi = 16$. The shift along the normal direction $d_{m,i} \in [-b_1, b_1] = \{-6, -4, -2, 0, 2, 4, 6\}$ pixels. The shift of orientation $\delta_{m,i} \in [-b_2, b_2] = \{-1, 0, 1\}$ angles out of $K = 15$ angles. So $|\Delta| = 21$. The orthogonality tolerance is $\zeta = .1$.

Experiment 1: Learning active basis. We apply the shared pursuit algorithm to a training set of $M = 37$ car images. The car images are 82×164. Figure (3) displays the results from the algorithm. The algorithm returns $n = 48$ elements using the stopping criterion (17). The first plot displays the learned active basis $\mathbf{B} = \{B_i, i = 1, ..., n\}$ where each B_i is represented symbolically by a bar at the same location with the same length and orientation as B_i. The intensity of the bar B_i is the average $\bar{r}_i$. For the remaining M pairs of plots, the left plot shows $\mathbf{I}_m$, and the right plot shows $\mathbf{B}_m = (B_{m,i}, i = 1, ..., n)$. The intensity of each $B_{m,i}$ is $r_{m,i}^{1/2}$. These M examples are arranged in descending order by their log-likelihood scores (16). All the examples with non-typical poses are in the lower end. We obtained similar result using active correlation. The examples displayed in figure (1) are produced after we force the algorithm to select 60 elements.

Experiment 2: Find and sketch. Using the learned model in experiment 1, we can find the car in the testing image shown in figure (4). The upper left plot is the testing image. The upper right plot displays the sketch of the car at the maximum likelihood scale and location. The lower left plot displays the maximum log-likelihood score over scale. The lower right plot displays the map of the log-likelihood at the optimal scale. We obtained similar result based on active correlation.

One issue that concerns us is normalization. In this experiment, we normalize within the whole image instead of normalizing within the sliding bounding box. We also tried the latter normalization scheme. Active correlation still selects the correct scale. However, for log-likelihood, the correct scale is near a local maximum instead of the global maximum. Another issue revealed by more experiments is that the maximum likelihood position is not always the correct position. We shall investigate these issues in future work.

Experiment 3: ROC comparison. Figure (5) displays 12 of the 43 training examples paired with their $\mathbf{B}_m = (B_{m,i}, i = 1, ..., n)$, $n = 40$, obtained by maximum likelihood.

Figure (6.a) and (b) display the active bases $\mathbf{B} = (B_i, i = 1, ..., n)$, $n = 40$, selected by the shared pursuit, using log-likelihood and active correlation scoring respec-

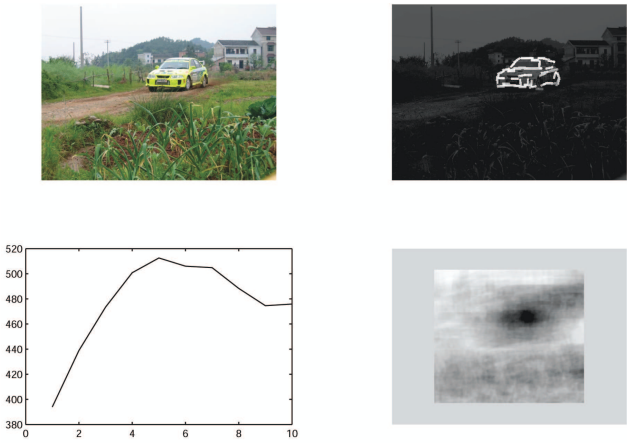

Figure 4. find and sketch. Lower left: the maximum log-likelihood score over scale. Lower right: the map of log-likelihood score at the optimal scale.

Figure 5. Some training examples and the corresponding $\mathbf{B}_m$.

tively. We also built an adaboost classifier [9] using the same set of training examples plus 157 negative examples, which are randomly cropped from natural scenes both with and without human figures, to represent enough diversity. The weak classifiers are obtained by thresholding the responses from the same dictionary of Gabor wavelets. Figure (6.c) displays the 80 Gabor elements selected by adaboost, where the red ones are those whose responses are greater than the corresponding selected thresholds, and the blue ones are otherwise.

(a) (b) (c)

Figure 6. (a) $\mathbf{B} = (B_i, i = 1, ..., 40)$ selected by active correlation. (b) $\mathbf{B}$ selected by log-likelihood. (c) 80 weak classifiers (from the same dictionary of Gabor wavelets) selected by adaboost. The red ones are the weak classifiers whose responses are larger than thresholds, while the blue ones are otherwise.

We then test on a separate data set with 88 positives and 474 negatives. Figure (7) displays the three ROC curves for active basis models learned by log-likelihood and active correlation, and the adaboost. The AUC (area under curve) for adaboost is .936. The AUC for log-likelihood scoring is .941. The AUC for active correlation scoring is .971. We did not implement cascade for adaboost. This example shows that our method is comparable to adaboost.

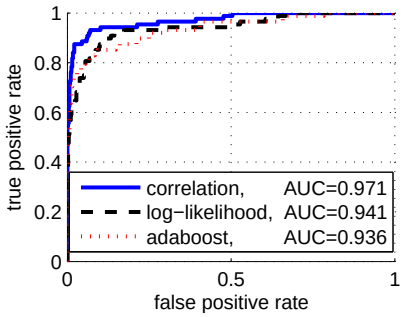

Figure 7. ROC curves for active basis models learned by active correlation and log-likelihood respectively, and adaboost. AUC means area under ROC curve.

Experiment 4: Mixture and EM. Suppose there are two categories in the training examples. We may assume a mixture model $p(r_{m,1}, ..., r_{m,n}) = \rho p^{(1)}(r_{m,1}, ..., r_{m,n}) + (1 - \rho)p^{(0)}(r_{m,1}, ..., r_{m,n})$, where $p^{(k)}(r_{m,1}, ..., r_{m,n}) = \prod_{i=1}^{n} \lambda_i^{(k)} \exp\{-\lambda_i^{(k)} r_{m,i}\}$, $k = 0, 1$. We can fit the model by the EM algorithm. Then we classify the examples into the two categories based on posterior probabilities produced by the last iteration of the E-step. After that, we re-learn the active basis model for each category separately.

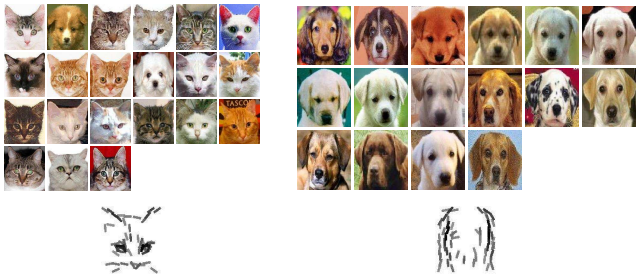

Figure 8. Top row: clustering result by EM. Bottom row: re-learned templates for the two clusters.

Figure (8) displays the 37 training examples. We first learn a common $\mathbf{B} = (B_i, i = 1, ..., n)$ with $n = 80$. Then we fit the mixture model on the coefficients of the 80 elements. The EM algorithm separates the examples into two clusters, as shown in figure (8), where there are 2 mistakes. Then we re-learn active basis models on the two clusters separately, with $n = 60$. The bottom row of figure (8) displays the learned templates. We can also re-learn the active basis models within the M-step in each iteration.

Experiment 5: Find and learn. By combining the codes in the first two experiments, our method has the potential to handle training images that are not aligned, as suggested by the following preliminary experiment. There are five images of cats that are of the same size but at different locations. The only supervision is to give the bounding box for

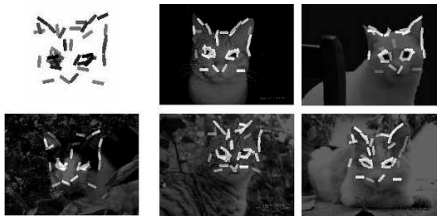

Figure 9. Find and learn. The first plot is the learned active basis. The rest of the plots sketch the identified cat faces.

the first image. We then fit the model on this single image, and use it to find the cats in the other images. Then we re-learn the model, and re-find the cats using the re-learned model. Figure (9) shows the results after 3 iterations, where the first plot is $\mathbf{B} = (B_i, i = 1, ..., n)$, $n = 40$.

Reproducibility: Data and source codes can be downloaded from the webpage listed on the title page.

6. Discussion

Residual image and inhibitive features. After activating basis $(B_{m,i}, i = 1, ..., n)$ and model $(r_{m,i}, i = 1, ..., n)$, we can also pool the texture statistics on the residual image that is not covered by $(B_{m,i}, i = 1, ..., n)$, and tilt $q(\mathbf{I}_m)$ on residual image. Along the same theme, we can also introduce inhibitive features on the residual image.

Center versus boundary. Even though one may view model (12) from a classification perspective, the model is trained by maximizing likelihood, which targets the center of the data, instead of the classification boundary. The advantage of targeting the center is that it is more efficient for small training samples, and more convenient for unsupervised learning.

Maximum entropy or minimum divergence. Model (12) is a special computable case of maximum entropy or minimum divergence principle [2], which tilts $q(\mathbf{I}_m)$ to $p(\mathbf{I}_m) = \exp\{\langle \lambda, H(\mathbf{I}_m)\}q(\mathbf{I}_m)/Z(\lambda)$, for some statistics $H()$, where $Z(\lambda)$ is normalizing constant. If H is the histogram of $(r_{m,i}, i = 1, ..., n)$, we get model (12) for shapes. If H consists of spatially pooled histograms, we get the Markov random field model [12] for textures.

Appendix

Proof of Theorem 1 Let $\bar{\mathbf{B}}_m$ be the matrix whose columns are orthogonal to the columns of $\mathbf{B}_m$, so that $\mathbf{B}'_m\bar{\mathbf{B}}_m = 0$. We can write $\epsilon_m = \mathbf{B}_m\tau_m + \bar{\mathbf{B}}_m\bar{\tau}_m$, where τ_m and $\bar{\tau}_m$ are n-dimensional and $|D| - n$ dimensional vectors respectively. Let $c_m = (c_{m,1}, ..., c_{m,n})'$, then $\mathbf{I}_m = \mathbf{B}_m(\tau_m + c_m) + \bar{\mathbf{B}}_m\bar{\tau}_m = \mathbf{B}_m\gamma_m + \bar{\mathbf{B}}_m\bar{\tau}_m$, where $\gamma_m = c_m + \tau_m$.

Under assumption (10), τ_m and $\bar{\tau}_m$ are independent because of the orthogonality between $\mathbf{B}_m$ and $\bar{\mathbf{B}}_m$, so $q(\tau_m, \bar{\tau}_m) = q(\tau_m)q(\bar{\tau}_m)$. Because of assumption (11), $\gamma_m = c_m + \tau_m$ and $\bar{\tau}_m$ are also independent, so $p(\gamma_m, \bar{\tau}_m) = p(\gamma_m)q(\bar{\tau}_m)$, where $p(\gamma_m) = \int g(\gamma_m - \tau_m)q(\tau_m)d\tau_m$ is the convolution of $g(c_m)$ with the Gaussian noise τ_m.

Under the linear mapping $\mathbf{I}_m = (\mathbf{B}_m, \bar{\mathbf{B}}_m)(\gamma'_m, \bar{\tau}'_m)'$, $p(\mathbf{I}_m)/q(\mathbf{I}_m) = p(\gamma_m, \bar{\tau}_m)/q(\gamma_m, \bar{\tau}_m) = p(\gamma_m)/q(\gamma_m)$ because the Jacobian terms get canceled. Let $r_m = \mathbf{B}'_m\mathbf{I}_m = (r_{m,1}, ..., r_{m,n})'$. Then under the mapping $r_m = \mathbf{B}'_m\mathbf{B}_m\gamma_m$, $p(\gamma_m)/q(\gamma_m) = p(r_m)/q(r_m)$, because again the Jacobian terms are canceled. So $p(\mathbf{I}_m)/q(\mathbf{I}_m) = p(r_m)/q(r_m)$. □

Acknowledgement

We thank the area chair and the three reviewers for their criticisms that help improve the presentation of the paper. We are grateful to Zhuowen Tu for helpful discussions. The work is supported by NSF-DMS 0707055, NSF-IIS 0713652, and ONR N00014-05-01-0543.

References

[1] TF Cootes, GJ Edwards and CJ Taylor, "Active appearance models," *PAMI*, 23, 681-685, 2001.

[2] S Della Pietra, V Della Pietra, and J Lafferty, "Inducing features of random fields," *IEEE PAMI*, 19, 380-393, 1997.

[3] JH Friedman, "Exploratory projection pursuit," *Journal of the American Statistical Association*, 82, 249-266, 1987.

[4] M Kass, A Witkin, and D Terzopoulos. "Snakes: Active contour models," *IJCV*, 1987.

[5] S Mallat and Z Zhang, "Matching pursuit in a time-frequency dictionary," *IEEE Signal Processing*, 41, 3397-415, 1993.

[6] J Mutch and DG Lowe, "Multiclass object recognition with sparse, localized features," *CVPR*, 11-18, 2006.

[7] BA Olshausen and DJ Field, "Emergence of simple-cell receptive field properties by learning a sparse code for natural images," *Nature*, 381, 607-609, 1996.

[8] M Riesenhuber and T Poggio, "Hierarchical models of object recognition in cortex," *Nature Neuroscience*, 2 , 1019-1025, 1999.

[9] PA Viola and MJ Jones, "Robust real-time face detection," *IJCV*, 57, 137-154, 2004.

[10] AL Yuille, PW Hallinan, and DS Cohen, "Feature extraction from faces using deformable templates," *IJCV*, 8, 99-111, 1992.

[11] SC Zhu, CE Guo, YZ Wang, and ZJ Xu, "What are textons," *IJCV*, 62,121-143, 2005.

[12] SC Zhu, YN Wu, and D Mumford, "Minimax entropy principle and its applications in texture modeling," *Neural Computation*, 9, 1627-1660, 1997.